2002

Algorithmic composition in contrasting musical styles

Tristan McAuley
Edith Cowan University

Edith Cowan University

Copyright Warning

USE OF THESIS

The Use of Thesis statement is not included in this version of the thesis.

Algorithmic Composition
in
Contrasting Musical Styles

This report is submitted as partial fulfilment
of the requirements for Honours in the
Bachelor of Science (Computer Science) :
Edith Cowan University
2002

Author: Tristan McAuley
Supervisor: Dr Philip Hingston

I certify that this thesis does not, to the best of my knowledge and belief:

(i) incorporate without acknowledgment any material previously submitted for a degree or diploma in any institution of higher education;

(ii) contain any material previously published or written by another person except where due reference is made in the text; or

(iii) contain any defamatory material

Acknowledgements

Thanks to my honours supervisor, Dr. Philip Hingston, for his patience and guidance throughout this year.

TABLE OF CONTENTS

1.0 - INTRODUCTION **1**

1.1 Interest 1

1.2 Research Questions 1

2.0 - LITERATURE REVIEW **4**

2.1 Background 4

2.2 Implementing Global Structure 5

2.3 Learning Local Characteristics Through Grammars 7

2.4 Modification of Note Sequences to Represent Improvisation 8

2.5 Evaluation of Generated Music 10

2.6 Conclusion 12

3.0 - METHOD **14**

3.1 Process Overview 14

3.2 Obtaining Source Data 16

3.2.1 Formatting the Sample Songs 16

3.2.2 Converting from MIDI 18

3.2.3 Formatting the Note Attribute Values 19

3.2.4 Obtaining and Storing the Required Data 20

3.3 Generating Note Sequences 21

3.3.1 Finite State Automata 22

3.3.2 Note Attribute Retrieval 26

3.3.3 The Generation of Note Sequences 28

3.3.4 Bossa Nova 29

3.3.5 Trance 30

3.4 Note Sequence Modification Techniques 30

3.4.1 Storage and Access 31

3.4.2 Bossa Nova 31

3.4.3 Trance 33

3.5 Instilling Global Characteristics 34

3.5.1 Activity table 35

3.5.2 Rule Sequence 36

3.5.3 Genetic Algorithm 38

3.5.4 Expanding the Solution 42

3.6 MIDI File Rendering 45

3.6.1 Chord sequence 46

3.6.2 Note Sequence to MIDI Conversion 47

3.6.3 Instrument Activity Table Traversal 49

3.6.4 Bossa Nova 50

3.6.5 Trance 52

3.7 Obtaining an Objective Critique 52

3.7.1 Questionnaire design 53

3.7.2 Song Generation and Preparation 56
3.7.3 Survey Procedure 56

4.0 - RESULTS 58
4.1 Introduction 58
4.2 Automated Composition of Songs in Two Contrasting Genres 58
4.3 Compositional Aid for Musicians 62
4.4 Generation and Modification of Note Sequences 63
4.4.1 Chord Voicing 63
4.4.2 Rhythmic Properties 64
4.4.3 Musical Phrasing 65
4.4.4 Instrument Conformance 66
4.5 Instillation of Global Characteristics 67
4.5.1 Form 67
4.5.2 Instrument Interaction 68
4.6 General 69
4.7 Summary 70

5.0 - DISCUSSION 72
5.1 Obtaining an Objective Critique 72
5.2 Note Sequence Generation and Modification 73
5.3 Global Characteristics 74
5.4 Fluidity 76
5.5 Future Considerations 77
5.6 Accomplishment of Objectives 79

6.0 - CONCLUSION 83

7.0 - REFERENCES 86

8.0 - APPENDIX A: DEFINITION OF TERMS 88
8.1 Technical 88
8.2 Musical 88
8.3 Internal 89

9.0 - APPENDIX B: SURVEY QUESTIONNAIRE 90

10.0 - APPENDIX C: SURVEY QUESTIONNAIRE 91

11.0 - APPENDIX D: OBJECT ORIENTED OVERVIEW 92

ABSTRACT

The aim of this project is to create software that will give non-musical people the ability to compose convincingly 'real' music in specific musical genres. By 'real' music we mean music which is not obviously "machine generated", recognisable as being of the selected genre, of an aesthetically pleasing quality (a very subjective concept) and usable in a commercial context.

To achieve this goal various computational techniques are used including genetic algorithms and finite state automata. The process involves an original, top down approach and a bottom up approach based on previous studies. A novel, skeletal structure is formed to depict a high level description of the song. Sequences of notes are generated based on sample note sequences taken from songs of the desired style.

This study involves generating songs from two different musical genres. Resulting compositions have been objectively assessed by student musicians. To facilitate this the resulting songs were translated into a machine-readable format. Instruments assigned to audition the generated songs were selected from a static, predefined list.

1.0 - INTRODUCTION

1.1 Interest

The intention of this study is to provide non-musically skilled people with the ability to compose songs in a desired genre. Currently there are few alternatives to hiring a costly, human composer when acquiring quality music for use in short films, computer games or multimedia presentations. Musicians may also benefit from this study through the development of automated compositional aids.

To achieve the said aim, music composed must exhibit the following qualities; seemingly of human origin (i.e. not obviously "machine generated"), recognisable as belonging to the desired genre and aesthetically pleasing (a very subjective concept). Further, generated compositions are required to be valuable in a commercial context.

The following processes are involved in the achievement of automated composition. Note sequences were generated using finite state automata. Finite state automata were modelled on sample note attribute sequences obtained from the genre. The independent combination of the generated note attribute sequences formed note sequences which were performed by each instrument.

Instillation of the global structure of a song is achieved through the evolution of an instrument activity table. A solution instrument activity table, representing instrument interaction typical to the genre, is evolved via a genetic algorithm. This table orchestrates the integration of the local and global characteristics of the song. That is, the arrangement of the generated note sequences and the triggering of note sequence modification techniques.

Further, the objective evaluation of the resulting songs has been accomplished through the gathering of feedback from student musicians. A survey was conducted to obtain this feedback. During the survey a number of consecutively generated songs were auditioned and participants were required to complete questionnaires. The questionnaires were designed to generate objective feedback evaluating the degree of success achieved in accomplishing the stated objectives and research questions.

1.2 Research Questions

- *Can a skeletal structure, representing typical instrument activity in a musical genre, be evolved using a genetic algorithm?*

Instilling global characteristics, such as instrument interaction and form has had little previous success. The novel application of an instrument activity table endeavours to embody global characteristics common to the desired genre. This is believed possible as common characteristics, from the trance and jazz genre, can be described in terms of instrument activity.

For example, it is common for the bass guitar and drums to be played constantly throughout a typical jazz song. This can be represented in terms of the song's structure, as a series of indicators within a table. Each row within the table, represents an instrument and each column a set time length, termed phase. An assumption is made that the length of the phase, two bars, will provide adequate precision in accurately modelling instrument activity within both genres.

- *Can sequences of notes, generated by finite state automata, reflect the characteristics found in the sample data used to create the finite state automata?*

Finite state automata are used to model the local characteristics contained within note sequences sampled from the genre. This technique has been used in previous studies achieving moderate success. The approach is unique in two ways; merging criteria utilised by the finite state automata; and independent generation, and combination, of each note attribute to form a complete note sequence.

That is, separate finite state automaton will be trained for each note attribute, pitch; position; duration; velocity. The assumption is that note attribute sequences, independently generated from the sample data, may be combined to produce note sequences representative of this sample data.

- *Can changes to sequences of notes through morphing, mutation, regeneration, crossover, or a combination of these methods, realise variations in note sequences typical to the genre selected?*

The objective is the development of the focal melody line, or improvisation, in a way that effectively models the genre. This is achieved through the use of a combination of the above note sequence modification techniques. The resulting, combined note sequences will form a melodic phrase that spans multiple phases.

It is anticipated that the process of modelling spontaneous, human improvisations on melodies, integral to the jazz genre, will be difficult. Modelling a specific musician's style, although possibly more feasible, will not be attempted. Instead the aim is to model a specific genre. A 'trial and error' type approach, with

feedback from qualified musicians, will be necessary to uncover an appropriate technique or combination of techniques.

2.0 - LITERATURE REVIEW

2.1 Background

Algorithmic musical composition is a relatively recent field of research. Results from this research vary from the generation of synthesised sounds through to the composition of whole songs. The different areas of research are often categorised by their objectives. For example, the goal may be automated composition with little human intervention or creating a tool for aiding a musician in his/her composition.

The interest in algorithmic control of music dates back "at least as far as the famous story of Musikalisches Würfelspiel (Dice Music)" (Alpern, 1995). This is a frequently referenced case in which short phrases, composed by Wolfgang Amadeus Mozart, may be played in any order. The choice of this order was decided from the rolling of a die.

With the introduction of computers more elaborate projects were undertaken. Lejaren Hiller and Leonard Isaacson are credited with writing the first algorithmic composition on a computer in 1957. This was called 'Illiac Suite for String Quartet' and was generated using "the rules of counterpoint, and analysis of existing works stored in a Markov matrix" (Brown, n.d.).

Since then different approaches have produced many different ideas and techniques in algorithmic composition. The algorithmic compositional process can be broadly classified into four approaches; a rule-based approach, a probabilistic approach, a connectionist approach, and an evolutionary approach.

A rule based approach is one which applies user specified rules to guide the compositional process. "These systems rely on the 'wisdom' of those who design the rules." (Brown, n.d.). A probabilistic approach presents a number of weighted choices for each consecutive note. This allows a more accurate modelling of the pieces analysed although note order is still relatively uncontrolled. An extension to this is a statistical analysis which also takes into account the local context of the melodic sequence, such as a Markov model.

The connectionist approach is modelled on the human brain and comprises of a web of interconnected nodes. This system is able to 'learn' patterns in music yet maintain a sense of local context. Extensive research in a variety of applications of this technique may be found.

An evolutionist approach uses "general tools for search and optimisation inspired by the biological theory of natural selection" (Pearce, 2000). A population, consisting of computerised representations of the music, undergo an evolutionary algorithm. Genetic operators such as crossover and mutation are applied to the population to evolve a solution. Each of the described approaches has its associated benefits and weaknesses, making it advantageous to use multiple methods in conjunction.

The intent behind algorithmic composition has been to provide automated tools for transforming, refining or extending musical melodies, for modelling composers and musical styles, or for generating original phrases of music to provide stimulus for new ideas. Algorithmic composition has also been devised to generate complete songs with little or no user input. This area of research has had little success with the global structuring of a song. Conclusions presented are subjective and varied due to inadequate and biased evaluation of the results.

2.2 Implementing Global Structure

Research in algorithmically composing music has attained little success in implementing a global structure within the song. Summaries of algorithmic techniques often state the sentiment that the "problem with many compositional algorithms is the lack of macrostructure" (Järveläinen, 2000). Discussion of research results often declares that the "most serious challenge, however, is global structure" (Chen, 2001). Any process attempting to model a genre must display macro characteristics as well as localised ones. For example, a description of Jazz during the 1950's will include distinguishing, global characteristics such as "improvising upon a fixed form and harmonic structure" (Morangelli, 1999).

Research that has focused on defining and implementing a global structure include COMPOzE (Henz, n.d.) and Contour (Alpern, 1995). The goal of COMPOzE is to "derive four-voice music pieces from given musical plans" (Henz, n.d.). A musical plan describes the harmonic progression of a song. It also contains predefined, descriptive names that detail dynamics the melody will comply to.

As with many research pieces in this field, the evaluation of the resulting work is vague and subjective. The conclusion states "resulting compositions are simplistic in style due to the rigid dynamic structure, but very pleasing from an harmonic and melodic point of view" (Henz, n.d.).

Musical plans, created in COMPOzE, are limited to guiding the dynamics of the melody and the chord progression. This method is not sufficiently complex or flexible to enable an accurate representation of global structures in two contrasting musical genres.

Contour "generates new melodies by 'expanding' existing melodies based on an analysis of their contour" (Alpern, 1995). An *overall* contour is used to indicate whether a melodic phrase rises or falls. This is done by noting whether each following note is lower or higher in pitch. If the number of rises is greater than the number of falls the phrase is labelled as a 'rise'.

Also used is a *meta* contour. This works in the same way as the *overall* contour but is applied to a sequence of *overall* contours. The author claims that compositions have a "sense of going somewhere, rather than merely wandering aimlessly", but states that the approach involves little analysis and is not very sophisticated. This lack of sophistication makes it unsuitable for modelling or analysing global characteristics of a specific musical genre.

Observations and analysis of musical genres are often in terms of instrument activity. For example, the following phrases describe characteristics of a Jazz sub-genre labelled the Cool School: "Bass and guitar become solo instruments in their own right"; "The Bass Drum was used infrequently but accurately to propel the soloist"; "The Bass assumed the responsibility for defining the pulse" (Morangelli, 1999). Based on this premise, it seems feasible to instill global characteristics within a composition through the use of an instrument activity table. The table will define which instrument is active during each stage in the song.

The formation of this table must reflect the intended genre. The choice of the method used by the 'critic' to evaluate each member is essential in evolving a viable solution. Previous studies using a neural network based critique of a population show little promise, "These partial failures of the system to achieve the stated aims were attributed to shortcomings of the data used to train the ANN critic" (Pearce, 2000). The difficulty in obtaining sufficient training data for a neural net further detracts from its viability.

A rule-based critic is more desirable. Rules can be extracted from the theory and analysis associated with the intended genre. Training data will not be needed for this approach. The general feedback from cases researched was that rule-based critics fared better than an ANN approach.

2.3 Learning Local Characteristics Through Grammars

A weighted finite state automaton (WFSA) is used by the 'Viterbi Jazz Improvisator' to "find a concatenation of melody fragments that best matches a given sequence of chords" (Hirzel, 2000). Input to the WFSA consists of user selected *patterns*, a melody of one bar length, and a user defined list of *canonical successors* for *patterns*. These define which patterns must follow each other. Transition probabilities between states that are not part of the list are calculated based on the pitches of the notes that must be adjoined.

The user is required to specify a chord sequence to which the Viterbi Jazz Improvisator will supply an appropriate *pattern* sequence. The primary aim of the Viterbi Jazz Improvisator is education, as indicated by the evaluation questions: "How easy is was it to play?"; "How useful is it as a practise?"; "How pleasant is it to the ear?" The answer to the last question, most relevant to this study, described the generated music as technically correct and pleasant to the ear, yet also monotonous, due to the unchanging *patterns*, indicating a need for a modification of the melody over time. This is discussed in the following section.

A paper titled "Probabilistic Grammars for Music" (Bod, 2001) investigates the use of probabilistic grammars in categorising and analysing the Essen Folksongs musical style. Results showed that a DOP-Markov model based on a history of three notes "can correctly predict up to 85.9% of the phrases for a test set of 1,000 folksongs" (Bod, 2001). This is an encouraging result for the use of a grammar-based technique in modeling a musical style.

Further research into using Markov models to learn musical styles has been done by Bradley J. Clement. His work studies the ability of a Markov model to learn harmonic progressions in different musical styles. Through the results of his experiments and the evaluation of other relevant studies, he forms the opinion that regular grammars are only adequate for representing "lower-level characteristics of music, such as harmonic progressions" (Clement, 1998). The characteristics this study aims to model (pitch, duration, rhythm and velocity) are considered by B. Clement to be lower level and thus feasible to be modelled by a regular grammar style approach.

Following the suggestions of Clement, this study uses independent finite state automata to learn each lower level characteristic. This assumes lower level characteristics, generated independently, can be combined to form new melodic sequences representative of the original sample data. Or, as put by Clement (1998) "assuming these properties can be learnt independently, the real challenge is to learn how they depend on each other to give a full representation of musical style". The presence of dependence between lower level characteristics is not investigated in this study.

2.4 Modification of Note Sequences to Represent Improvisation

A melody that is not modified or developed over time can produce an undesired monotony in the music. This was pointed out by the author's evaluation of the Viterbi Jazz Improvisator discussed earlier. In addition, a compositional process that does not vary the melody will be unable to accurately model a musical style or genre.
The following paragraphs relate studies involving modifications and improvisations of melodies and drumbeats.

Vox Populi, created by Manzolli, Moroni, Von Zuben and Gudwin, utilises an evolutionary process to evolve a population of 'choirs'. Each choir is a group of four voices; soprano, contralto, tenor and bass. A rule-based process is used to evaluate each member of the population on their melodic and harmonic content. The "best individual" from each generation is selected and played, producing a changing, evolving melody.

This technique uses a rule-based critic to instil desired musical properties. Although the aim of Vox Populi is not to model a specific musical genre, a complex, genre specific rule set will generate modified melodic sequences that are more in line with the desired genre.

"Algorithmic Composition Methods for Breakbeat Science" is a paper by N. Collins which addresses the problem of monotony in beats found within the dance musical genre. "The goal is to create an output beat in a consistent style with continuous subtle variation rather than stale repetition" (Collins, 2000). The style chosen is 'UK Garage', a sub-genre of Dance. Two methods are used to modify the beat.

The first method involves *probability templates*. These model the probability of a percussion instrument occurring in a particular time-slot. The probabilities must restrict the generated beat from being too 'busy', but still allow enough variation.

The second method involves a collection of *cells*; each cell is a bit sequence in which a '1' represents activity for the instrument at that time-slot. Cells are selected according to their associated probability and combined to produce a sequence of varying drumbeats. Combining cells may produce an unconventional breakbeat. Filters are used to remove "exemplars of bad style" (Collins, 2000).

This study discovered that a more formal structure between beats is needed to accurately model the 'UK Garage' genre. The probabilities used in the first method were assigned independently of each other. However, dependence between successive probability templates was considered necessary. The second method was also considered too unstructured. A structured beat pattern was proposed to further the resemblance to the 'UK Garage' genre.

"Improvised Music with Swarms" (Blackwell, 2002) claims to be the first to apply the particle swarm algorithm to music. SWARMMUSIC focuses on interacting with a human in real time to produce freely improvised music (improvisation that does not adhere to conventional musical constraints). Instead improvisation is induced by "change in a musical landscape" (Blackwell, 2002) and thus does not need extensive, hardwired musical knowledge.

The author claims to be very successful in emulating a human, writing "it is hard to believe that this is not of human origin" (Blackwell, 2002). The author indicates a similar success in the interaction between SWARMMUSIC and a human performer. However, no objective or quantitative data is presented. The author does comment that "musical structure is an emergent property of the musical swarm" (Blackwell, 2002).

Although the results seem impressive, this technique is unsuited to the process of modeling a musical genre. Improvisation is achieved through interaction with a live musician which is not relevant to the objectives of this project. Freely improvised music is a style of music within itself. To emulate the selected Jazz subgenre improvisation must conform to musical constraints typical of that subgenre. It would not be feasible to impose a structure on this inherently formless, improvisational process.

Papers containing a detailed analysis of the genre in question can provide style specific rules useful in guiding the process or evaluating results. In "A History of Jazz" the melody, harmony, tonality, rhythm, textures and forms of a particular age are described in detail. From this descriptive analysis it is possible to derive rules to which melodic variations must adhere. For example, "The 20th Century composer had little

use for standard patterns - in phrase or repetition of theme or motif" (Morangelli, 1999) implies a substantial amount of variation is needed between melodies.

Formal theory illustrating the method behind improvisation or composition may also provide a guide in devising or evaluating melody modification methods. The online tutorial 'Jazz Improvisation Course' discusses the different approaches to improvisation. These are "embellishments of the song's melody", "focus on chord tones" or "the underlying scale" (Furstner, 2000).

Another method of evaluating the degree of success achieved by a method is to obtain feedback from a professional musician, practised in the relevant genre. A further test between the modified melody and the sample data can verify the validity of this melody to the genre. However, remember that a melody accurate to the genre will not always imply a valid solution within the context of the current composition.

2.5 Evaluation of Generated Music

A problem, often noted, with research in the area of algorithmically composed music is the lack of objective and thorough evaluation of results obtained.
G. Papadopoulos and G. Wiggins (1999) express this lack of methodology in their paper "AI Methods for Algorithmic Composition: A Survey, a Critical View and Future Prospects". They highlight two problems: first, there is usually no evaluation of the output by real experts (e.g. professional musicians) in most of the systems, and second, the evaluation of the system (algorithm) is given relatively small consideration with respect to the length of the report.

A study by Marcus Pearce, 'Generating Rhythmic Patterns: a Combined Neural and Evolutionary Approach' (2000), has attempted "to evaluate in more objective terms whether the system fulfills the specified aims". To do this the following three experiments were conducted: a musical Turing test, style classification by human subjects and judgements of diversity by human subjects. In addition to these experiments, the system was evaluated by a musician and informally judged on usability and practicality.

Experiment one, the musical Turing test, mixed a number of system-generated and human-generated samples. Subjects were then asked to distinguish between the two sample types. This Turing style test was expected to "factor aesthetic judgement out of the equation" (Pearce, 2000).

Results revealed that subjects were able to classify most system-generated samples, but were unable to classify human generated samples better than randomly. Speculation about these results suggested that a bias was present towards classifying samples as system generated. This bias could possibly be due to the subjects' lack of familiarity in the genre or the criteria which the subjects used to distinguish between sample types.

Experiment two required subjects to evaluate whether each member of the same sample set belonged to the intended genre or another. "If the proportion of system generated patterns correctly classified according to style was equal to or greater than the proportion of human generated patterns correctly classified then the system generated patterns could be considered to be in the correct style" (Pearce, 2000). Results indicated a partial success, with the proportion of system generated samples correctly classified significantly greater than if chosen randomly, but fewer than the number of correctly classified human generated samples. Lack of knowledge of the musical style by the subjects was again speculated to have a negative affect on the results.

Experiment three was designed to evaluate the amount of musical variation in the samples generated. Results again showed that the subjects perceived the system-generated samples as having a lower degree of variation when compared to their perception of the human generated samples. An informal evaluation by a musician also provided valuable assessment regarding the systems usability and practicality.
This involved a survey containing general, open-ended questions about the musician's experiences with the system.

A further paper by M. Pearce and G. Wiggins (2001) focuses on constructing a framework from which machine compositions can be evaluated objectively. The authors quote: "Our concern in this paper is the evaluation of music composed by computer programs". Objective evaluation of an inherently subjective artwork is attempted through empirical experimentation. The "framework involves four components: specifying the compositional aims; inducing a critic from a set of example musical phrases; composing music that satisfies the critic; and evaluating specific claims about the compositions in experiments using human subjects.

The first component states that the compositional aims of the system must be explicitly defined. The intent of modeling a composer, genre or generating a new style must be stated. The degree to which the genre will be adhered to and whether the intent is to compose complete musical pieces or compositional sub-components must be expressed.

A machine learning technique is recommended for inducing a critic from example musical phrases. The technique selected will depend upon the "musical domain" (Pearce, 2001). Consideration in example selection and the representation used must be taken to avoid bias. The selection of techniques and approaches used in the automated compositional process need to be stated and justified. Also, any musical, theoretical assumptions made must be explicitly declared.

The method of evaluation of compositions produced by the system is similar to that of a Turing test. As described above, human subjects must distinguish between system generated and human composed pieces. It is argued that this method of evaluation tests for any additional or absent features that may identify the piece as of non-human origin. "These features may be taken to include such elusive notions as aesthetic quality or perceivable creativity" (Pearce, 2001).

An evaluation of an example system using the described framework detected that the following issues needed addressing: Creating a "reliable and appropriate means of inducing a critic from a body of music" (Pearce & Wiggins, 2001) and effectively evaluating the degree of 'creativity' within a piece. The positive results of this study included the ability to make refutable, "scientific claims about the degree to which a system fulfills its compositional aims" (Pearce & Wiggins, 2001). This study aims to use a similar approach in obtaining quantifiable, objective evaluations of the approaches taken.

2.6 Conclusion

To address the challenge of instilling global structure within a song, an instrument activity table has been evolved. The instrument activity table depicts global characteristics, such as instrument interaction and form, by indicating which sections, within a song, an instrument is to be active. A genetic algorithm is utilised to produce a table representing instrument activity typical to the intended genre. A rule-based critic guides the search. The rules comprising the critic are derived from the analysis and observations of the musical genre.

Following the suggestions of Clement, independent finite state automata will learn the local characteristics of note attribute sequences displayed by sample data from the genre. The following assumption is made. Note attribute sequences, independently generated by finite state automata, may be combined to accurately represent the original, sample note sequences.

The combination of formal theory and feedback from a professional musician will be used to evaluate a number of varying, note sequences modification techniques. The failings, and achievements, of related studies will aid in attaining a capable technique. The method of activation, for the selected technique(s), will be integrated into the instrument activity table.

An objective evaluation of the compositional approaches, accomplishment of specified goals and aesthetical quality of the resulting pieces will be attempted. This evaluation will be in the form of a survey conducted with student musicians. A Turing style approach, discussed previously, is not feasible in the context of this study. The classification of a complete piece as machine-generated, or of human origin, will be dependent on the song's familiarity to the participant.

3.0 - METHOD

Sample data is manually selected from MIDI song files typical to the intended genre. Formatted versions of this data is used for populating the resulting song with note sequences. Owing to previous success in similar applications, finite state automata are used to model the local characteristics present in the data.

This study trials two unique variations to previous approaches taken. Firstly, note attributes pitch, position, duration and velocity, are independently modelled. Secondly, the merging of states within each finite state automaton is dependent on both the note attribute's value and the note attribute's position within its original sequence.

The global structure of a song is represented by an instrument activity table. This table orchestrates the generation and placement of note sequences within the song. A genetic algorithm is utilised to evolve a solution instrument activity table which represents instrument activity within the desired genre. To guide this search, a rule-based critic is used. Rules comprising this critic are derived from associated theory.

A survey, involving Jazz students, was conducted to obtain an accurate, objective appraisal of the system. This involved the translation of the generated songs into a machine-readable format enabling the auditioning of each song for evaluation. The format chosen was MIDI. This was due to its universality, portability and usability.

The following sections provide an overview of the algorithmic process and a description from an object oriented perspective. A data flow diagram is used to model the algorithmic process. Dashed lines depict the boundaries of each section addressed separately in the corresponding subheadings. The principle objects and their interaction with each other are explained.

3.1 Process Overview

An arbitrary number of MIDI files, representing local characteristics common to the intended genre, are supplied by the user. Based on these files, the algorithmic process produces the resulting MIDI file. The following diagram depicts a high level representation of this process.

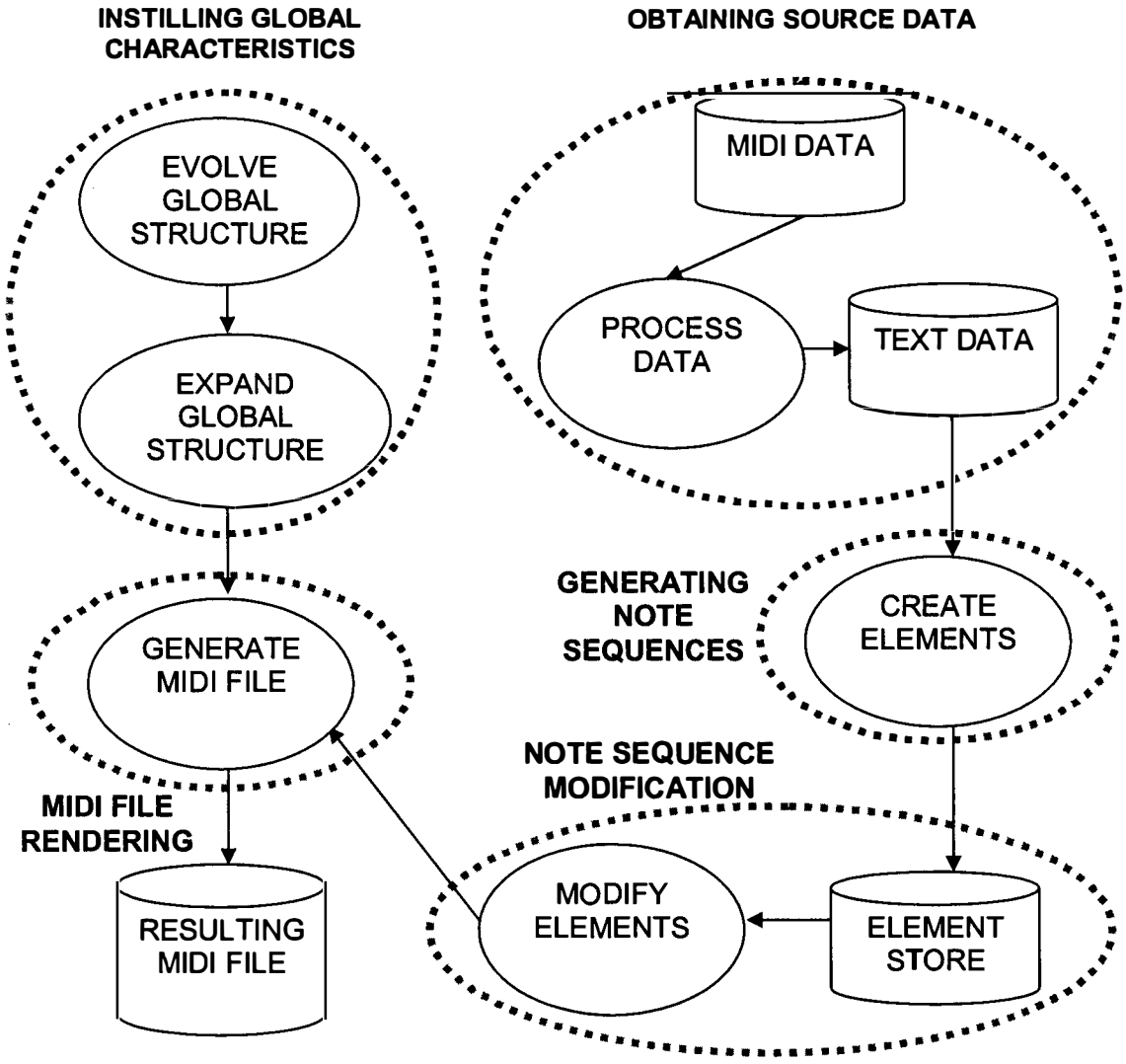

Figure 3.1: Level 1 Data Flow Diagram of the Algorithmic Process

"MIDI data" refers to preformatted MIDI files containing note sequences in the desired genre. The necessary data is extracted from these MIDI files and stored in text files. The information contained in these text files is used to create the finite state automata. Finite state automata provide each element with its base note sequence. An element is the internal representation of an instrument and is discussed in detail in a later section.

A genetic algorithm is used to evolve a global structure in the form of an instrument activity table, discussed earlier. The resulting table dictates the activity of instrument groups which must then be expanded to represent each individual instrument. This involves the linking of each cell to its corresponding element. The table also includes information about the occurrence and type of modifications to perform on each element. The song, represented by the expanded table, is then converted into the machine readable format of MIDI.

The dashed lines in Figure 3.1 depict the following logical partitions of the algorithmic process: obtaining source data; generating base note sequences; instilling global characteristics; note sequence modification techniques; midi file rendering. Each of these partitions is addressed individually in the subsequent sections.

3.2 Obtaining Source Data

Note sequences found in the generated song are constructed from sample song files supplied to the software. The sample songs are chosen to contain local characteristics inherent to the desired genre. The process applied to the data from these sample songs produce note sequences modelling each instrument's characteristics.

Referring to Figure 3.1, PROCESS DATA retrieves the required data from the pre-formatted MIDI (sample) files provided. This data is then formatted and separated into the appropriate text files. Finite state automata, discussed in a later section, are then constructed from these files to generate note attribute sequences. The following section details the steps in extracting the desired information from the sample songs provided.

3.2.1 Formatting the Sample Songs

Prior to running the software a number of sample song files must be selected and formatted. Songs chosen determine the style and genre of the song generated. MIDI representations of the songs selected may be located on the internet and/or bought from sample CD vendors. Approximately ten to twenty sample note attribute sequences are obtained for each instrument from the MIDI files used.

Once a suitable collection of MIDI files are obtained, each file must be manually formatted. This enables the software to extract the required information from each file and separate this information for each instrument. The formatting also provides for the logical comparison of information in different keys.

A graphical depiction of the format of an example MIDI file is shown in Figure 3.2. There are a number of tracks (in this example 12) and each track is assigned an instrument and a sequence of notes. For example, track 3 is played by a nylon guitar, 'Acoustic Guitar (nylon)'. The associated sequence of notes are displayed as black lines on a yellow background running horizontally along the track.

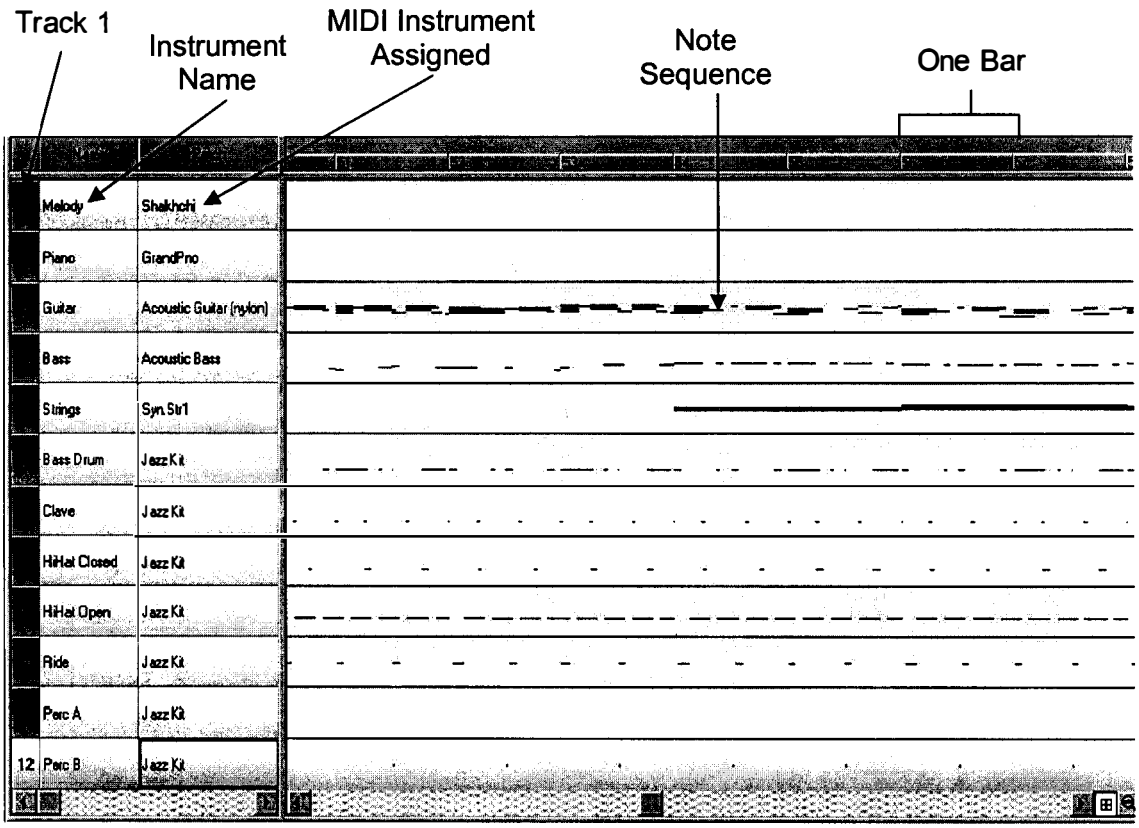

Figure 3.2: Graphical representation of a MIDI file as shown in Cakewalk ©

The base time unit this software deals with is two bars. Therefore, the MIDI file must be partitioned into two bar sections. This means each sequence of notes for each track must be divided into two bar segments. Each segment is then placed into a new track separate from the unformatted MIDI file.

A track name must be assigned to each segment indicating which instrument type these notes belong to. For example, a segment from the sequence of notes associated with the 'Acoustic Guitar (nylon)' will be assigned the identifier 'Guitar'. Percussive instruments are assigned an instrument type depending on the characteristics of the notes within the segment as well as the instrument originally designated to play that segment.

To allow a meaningful comparison between different segments, each segment must be transposed to the same key. The key chosen is C and the transposition of a segment into this key involves either raising or lowering each note a uniform distance - this distance being dependent on the key of that segment. The MIDI files containing the formatted data are then stored in a directory accessible by the software.

3.2.2 Converting from MIDI

Note sequences present in the formatted files must be converted from their MIDI representation into a format readable by the software. The software defines a note as the sum of its four attributes, pitch, position, duration and velocity. For each instrument type, these attributes are taken from the MIDI files and are stored in separate text files.

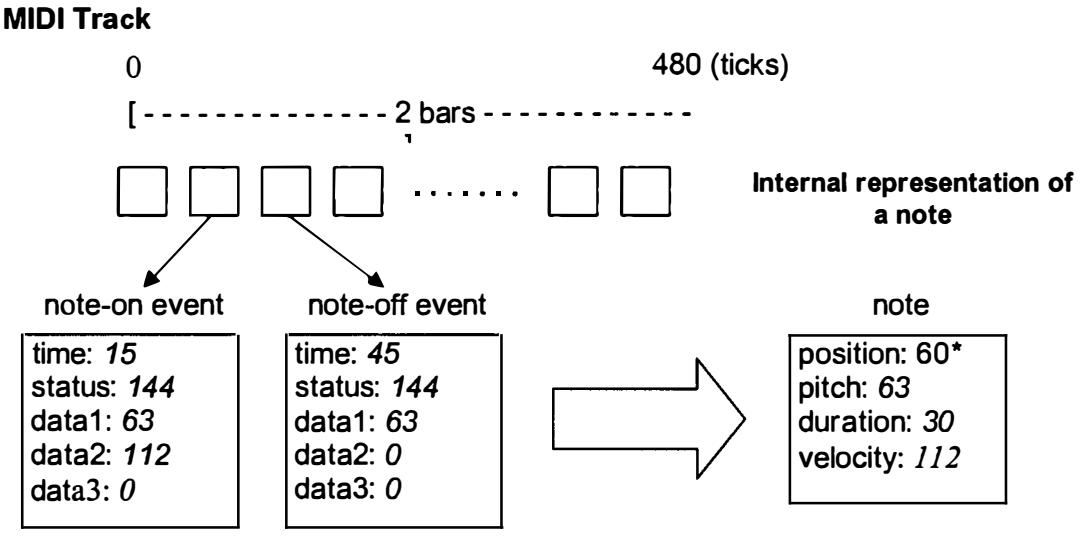

Figure 3.3: Graphical depiction of the conversion between the MIDI and internal representations

MIDI files represent a note sequence as a series of note-on and note-off MIDI events. A MIDI event contains the time since the last event, a status, three variables, data1, data2 and data3. The status indicates the type of MIDI event. A MIDI event with a status value between 127 and 160 indicates a note-on or note-off type MIDI event. These two event types are the only events this software manipulates.

The time value for an event specifies the time since the last event occurred. This measurement is in 'ticks' and is relative to the length of a bar and not a measure of time itself. To illustrate, a MIDI file with the setting '120 ticks per quarter-note' will contain a resolution of 120 ticks for each quarter of the bar (480 ticks per bar). The absolute time of an event is the time, in ticks, since the beginning of the track.

The first variable, data1, stores the pitch of the note. This is a value between 0 and 127 which corresponds to a note on the (musical) keyboard. For example, value 52 specifies the note E in the third octave. The variable data2 stores the velocity of the note. The velocity value is also between 0 and 127 and indicates how hard the note has been 'struck' (played). A smaller value indicates a softer sound and a value of 0 results in no sound.

Each note-on event has a corresponding note-off event. These events are linked by an identical pitch value. The duration of a note, in ticks, can be measured by subtracting the absolute time value of the note-off event from the absolute time value of the note-on event. A note-off is usually identified by the status value of the event but may also be defined similar to its corresponding note-on event but with a data2 (velocity) value of 0.

As mentioned previously, notes are represented internally by the software as consisting of four attributes; pitch, position, duration, velocity. The position attribute defines the note's position within the two bar segment. This value is in ticks where each segment contains 960 ticks (480 per bar). To facilitate the conversion of the internal representation back into MIDI, pitch, position and duration values are kept as their MIDI equivalents. The following pseudo code demonstrates the conversion process.

```
for each midiEvent in sequence                    // for each MIDI event in the track
        if (event = note-on) then
                // retrieve the pitch, velocity and position values from each event
                pitchValue[currentNote] := midiEvent.data1;
                // currentPosition holds the number of ticks occurred since the start
                positionValue[currentNote] := currentPosition + midiEvent.time;
                // update currentPosition
                currentPosition := currentPosition + midiEvent.time;
                velocityValue[currentNote] := midiEvent.data2;
        end if;
        if (event = note-off) then
                bool match := false;
                int count := 0;
                while (not match)        // find note belonging to note-off event
                        match := (pitchValue[currentNote-count] = midiEvent.data1)
                        count := count + 1;
                end while;
                // update currentPosition
                currentPositon := currentPositon + midiEvent.time;
                // determine the duration of the note matched
                durationValue[currentNote-(count-1)] :=
                                        currentPositon - positionValue[currentNote-(count-1)];
        end if;
end for;
```

Figure 3.4: Pseudo code demonstrating MIDI conversion algorithm

3.2.3 Formatting the Note Attribute Values

Before the converted note sequence is written to the appropriate files, each note attribute sequences must be formatted. The formatting serves two purposes. Firstly, the note position values of the sequences are offset so each two bar segment begins at tick 0. This enables the meaningful comparison of note positions from different segments.

Secondly, the position and duration values are rounded to the nearest 15 and the velocity values are rounded to the nearest 4. The effect of this is an increased merging of states in the finite state automata constructed, discussed in detail later. The benefit of formatting is an increased variability in generated note sequences. This reduces the amount of sample data required to generate original, varying note sequences.

When applied to position and duration attributes the number 15 corresponds to $1/32^{nd}$ of a bar; the musical term for a note of this duration is a semi-quaver. The two bar segment is conceptually partitioned into sections equal to the length of a semi-quaver. A note not beginning at the start of a section is shifted to the nearest section. Similarly, a note's duration is truncated or extended to comply with this formatting. The velocity values are either raised or lowered to the nearest value divisible by 4.

3.2.4 Obtaining and Storing the Required Data

The following paragraphs describe the process involved in obtaining the required data from the pre-formatted sample songs and storing them into the correct text files. Data is obtained with each run of the software and discarded before subsequent runs. The above sections describe in more detail the more complex algorithms involved in this process.

The name and path of each formatted MIDI file is taken from the directory specified by the genre selected. Each file, in turn, is accessed and each track this file contains loaded. The following steps are applied to each track.

Each note event in the track is converted to the internal software representation using the algorithm described in Figure 3.4. The four different note attribute sequences acquired are stored in separate arrays. All four arrays are formatted, as explained in the section 'formatting the note attribute values', before they are written to a text file.

The name of the current track is obtained. This name defines the instrument type the set of notes in the track belong to. The track name concatenated with the attribute name specifies the file name to which the note attribute array is appended.
A termination value '-1' and a new line character follows the array of note attribute values appended to the text file.

For example, an array of formatted, position values are derived from the sequence of notes contained in the current track with track name 'guitar'.
The destination file is determined by appending the track name, 'guitar', with the note attribute type, 'Positions'. The array of position values is therefore appended to the file 'guitarPositons.dat', the file extension '.dat' is a constant. A termination value of '-1' follows each sequence of note attributes. The diagram below shows sample note attribute sequences from the guitarPositions.dat text file.

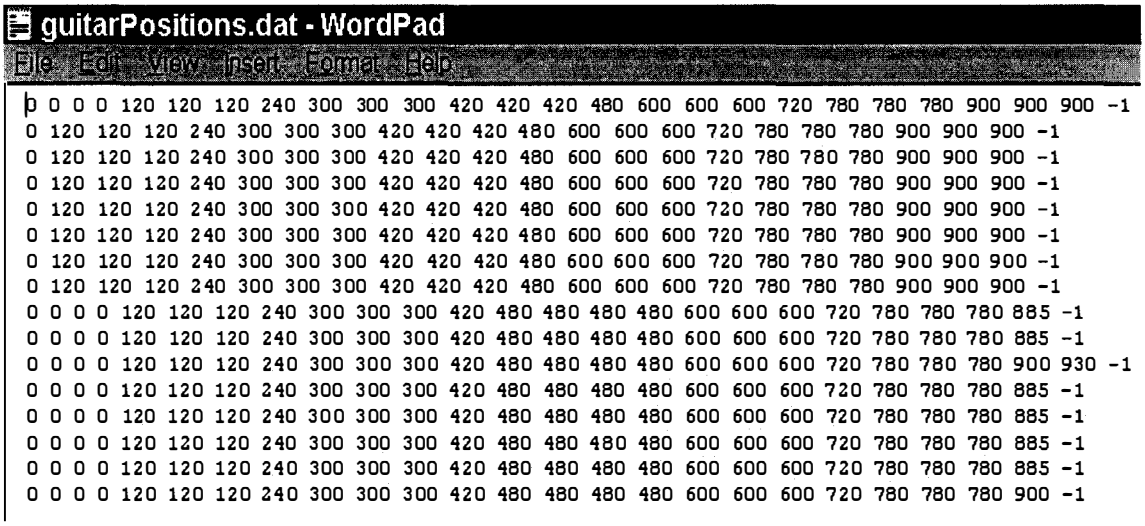

Figure 3.5: Note attribute sequences obtained from the sample MIDI files

3.3 Generating Note Sequences

Element objects facilitate the generation of musical notes that are assigned to an instrument when the midi file is rendered. An element object is named according to the instrument type it is intended to represent, such as guitar, bass drum or melody.
Each element object contains a sequence of notes equal in length to one phase. The time length of one phase is two bars, explained in the previous section 'Obtaining Source Data'.

A predefined set of instrument types is defined for each genre. An element object is created for each instrument type in the set. The name of an element corresponds to the name of the text files containing the data sequences for that instrument type. A sequence of notes is generated, based on the data contained in these associated text files, and stored in the element object. The technique used to generate the sequence of notes uses finite state automata.

Previous studies involving finite state automata have shown them to be effective in modelling the lower level characteristics of music. The unique approach adopted by this study is the modelling of each note attribute separately. An element is assigned a note sequence comprised of four, independently generated note attribute sequences.

3.3.1 Finite State Automata

A finite state automaton "is a mathematical model of systems which have a finite number of internal states and respond to external world just by changing their internal state." (Alberti, Marelli, & Sabadini, 1993). Finite state automata may be classed as deterministic or nondeterministic.

The transition performed by a nondeterministic finite state automaton may not be uniquely determined. For a given state and input, multiple, different transitions are possible. A mathematical definition is provided by Roberto Giovannetti (1993), "a nondeterministic finite automaton is a quintuple <Q,I,delta,q0,F>; where:

- Q is a finite set of states;
- I is a finite set of input symbols;
- delta is the, possibly partial, transition function

  delta: Q X I->P(Q), where P(Q) is the set of all subsets of Q
- q0 element of Q is called the initial state;
- F contained in Q is called the set of final states".

A finite state automaton may further be classed as being probabilistic. A probabilistic finite state automaton associates probabilities with each transition to a state. A finite state automaton is also categorised by its method of output. One that produces an output for each state is called a Moore machine and one that produces an output for each transition is called a Mealy machine.

The following paragraphs describe the unique logic used in study for the implementation of a probabilistic finite state automaton. This finite state automaton produces output for each state traversed and is thus classed as a Moore machine. An example is shown below in Figure 3.6.

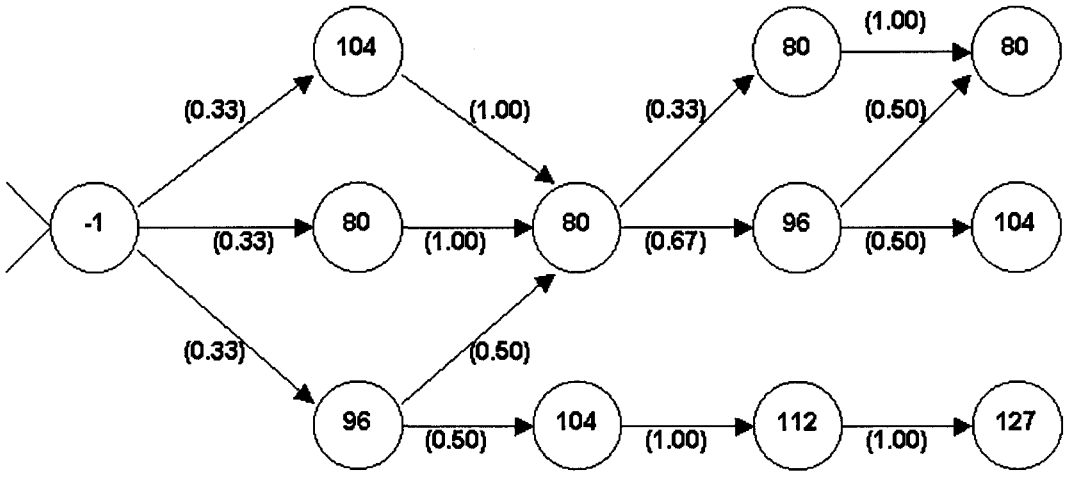

Figure 3.6 A probabilistic finite state automata

Finite state automata are implemented as an array of states. Each state contains the note attribute value and its position within the source, note attribute sequence. A state also contains a list of links to the array position of other states. Each link has an associated probability.

The algorithm, devised by this study for creating finite state automata, receives a number of note attribute sequences and fills the data structure described above. A pseudo code representation of this algorithm is presented in Figure 3.7.

The first position in the array of states is a start state. The start state is a dummy state with an special value for the note attribute and a position value of 0. The start state is the starting point for the algorithm when manipulating each data sequence.
This results in the start state containing a link to the first value of each note sequence entered. When retrieving values from the array of states the start state is the first state traversed.

The algorithm creating the finite state automaton create states, and links between these states, from the note attribute sequences supplied. Each note attribute is sequentially accessed and its position within that sequence determined. When a unique note attribute and position combination is encountered, the algorithm will add a new state to the 'Array of States'.

```
// create the dummy start state
create a startState at stateArray[0] with noteAttribute := -1 AND notePosition := 0;empty = 1;
for each noteAttribSeq provided                    // noteAttribSeq = note attribute sequence
        currentState = 0;
        for each position in noteAttribSeq         // for each note attribute in noteAttribSeq
                // search stateArray for any existing states with matching note attribute
                  and position values
                for search in 1..empty
                        sameState := -1;
                        if (stateArray[search].noteAttribute = noteAttribSeq[position] AND
                                stateArray[search].notePosition = position+1)
                                // if found, record position of matching state
                                sameState := search;
                        endif;
                endfor;
        // if matching state found, create a link to the matching state
        if (sameState>0)
                stateArray[currentState].createNewLink(sameState);
                currentState := sameState;         // update current state position in stateArray
        // if no matching state found, create a new state to store the noteAttribute
          in noteAttribSeq
        else
                stateArray[currentState].createNewLink(empty);
                create new state at stateArray[empty] with:
                                                   noteAttribute := noteAttribSeq[position] AND
                                                   notePosition := position+1;
                currentState := empty;             // update current state position in stateArray
                empty++;                           // update pointer to next empty array position
        endif;
endfor;
return empty;                                      // return the length of stateArray
```

Figure 3.7 Pseudo code demonstrating the formation of a finite state automaton

A link to this state is created in the current state. If the note attribute and position are not unique, the algorithm will create a link to the matching state and no new state will be created. The following diagrams illustrate an example of this process. The sample note attribute sequences processed are shown in Figure 3.8. Figure 3.9 depicts the status of the 'Array of States' after the first sequence and first value, 96, of the second sequence has been accessed.

The next value traversed is the non-unique combination of note attribute 80 and position *2*. A link to the matching state, 2, is made in the current state, 5. This is shown in Figure 3.10. Next, a unique combination of note attribute *96* and position *3* is found. A new state is created, 6, and a link to this state is made from the current state, 2. This is displayed in Figure 3.11.

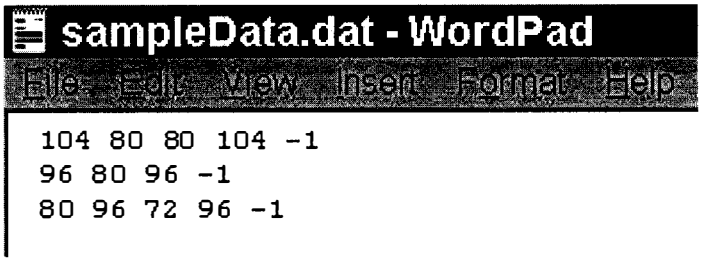

Figure 3.8 Sample note attribute sequences

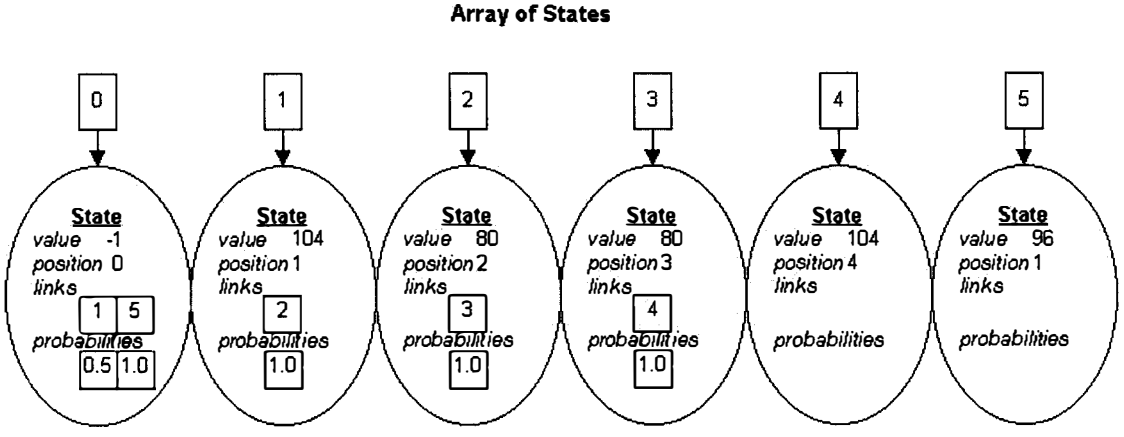

Figure 3.9 Internal representation of a finite state automaton

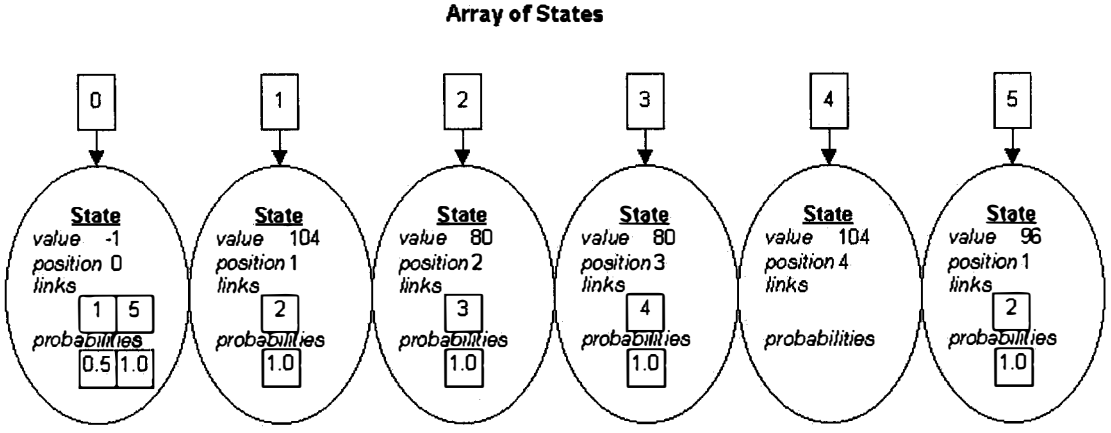

Figure 3.10 Internal representation of a finite state automaton

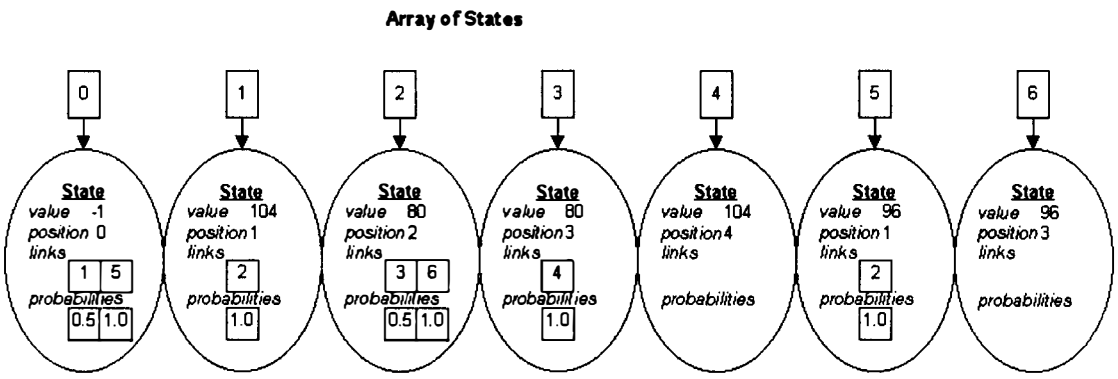

Figure 3.11 Internal representation of a finite state automaton

Once all note attribute sequences have been processed a probability is associated with each link in each state. The probability of a link is one divided by the sum of all links in the state. Multiple links to the same state are possible from within the one state.

The associated probability of traversing to the linked state is the number of these links divided by the sum of all the links. The probability assigned to a link includes the sum of all probabilities assigned to previous links in the state. This provides for the retrieval of the note attribute values from the data structure.

3.3.2 Note Attribute Retrieval

The construction of an element initiates the creation of four finite state automata. Note attribute sequences associated with the element are processed and stored in the data structure described above. New note attribute sequences are retrieved from the data structure and combined to produce the element's note sequence in the following way.

The first note attribute sequence retrieved is the position values. The number of position values retrieved determines the number of notes contained in the note sequence. The maximum length of a note sequence is two bars, or 960 ticks.

The following algorithm describes how position values are retrieved from the finite state automaton.

```
if (NOT stateArray.empty())                          // check data structure is not empty
        position := 0;                               // initialise position and count to 0
        count := 0;
        while                                        // begin while loop
                // retrieve number of links in the state at the current position
                numLinks := stateArray[position].getLinkCount();
                randomProb := random();              // generate a random probability
                // if number of links = 0, end of traversal path has been reached
                if (numLinks=0)
                        break;                       // exit while loop
                endif;
                // find the next link based on the random probability generated
                for each theLink in 1..numLinks
                        // compare the probability associated with link 'theLink' with the
                                random probability
                        if (stateArray[position].getProb(theLink)>randomProb)
                                // array position of next traversal state selected
                                newLink := stateArray[position].getLink(theLink);
                        endif;
                endfor;
                // set the current note's position value to the value retrieved from the stateArray
                noteSequence[count].setPosition(stateArray[newLink].getValue());
                count++;                             // update the note sequence index
                position := newLink;                 // update the current stateArray index
        endwhile;
        return count;                                // return the number of values retrieved
endif;
```

Figure 3.12 Pseudo code describing the retrieval of note attribute values from a finite state automaton

Other note attribute values, pitch, velocity and duration, can be retrieved using a similar method. The first value is retrieved by traversing the start state of the state array. A traversal is accomplished by randomly selecting the next state from the list of links contained in the current state and retrieving the next state's value. The number of traversals needed is equal to the number of positions retrieved.

A state containing no links indicates the end of a sample sequence of note attributes. A state with no links may be encountered before retrieving the number of pitches required. In this event, the next state is set to the second state selected. The second state is the state selected from the dummy, start state. The return to the first, selected note attribute provides a consistency for note attribute retrievals that reach an end state.

The following example, illustrating this process, refers to Figure 3.11. A note attribute sequence of length four is needed. The position is initially set to the first state in the array, state 0, and the random number 0.327 is generated. This value is less than 0.5 and results in a traversal to state 1 and the retrieval of the value 104.

The next traversal must be to state 2 as this is the only link. A value of 80 is retrieved. Next, a random number of 0.842 is generated. This value is larger than the probability associated with the first link and less than the associated probability of the second link. This means the second link is selected and a traversal is made to state 6. A value of 96 results from traversing to this state.

A further note attribute value is needed to complete the sequence, but no links are held by state 6. The position is reset to the second state selected, this was state 1. The traversal to this state results in the retrieval of the value 104, thus fulfilling the required note attribute length of four.

3.3.3 The Generation of Note Sequences

Associated with each element is four text files holding the four different types of note attribute data. This data is the sample data obtained for the instrument type the element represents. For each element, four separate finite state automata are created, one for each note attribute; position, pitch, duration, and velocity.

For example, note attribute sequences found in the text file 'bassPitches.dat' are used to create a finite state automaton for the bass element. See above for a detailed explanation on how finite state automata are created. The resulting finite state automaton will model bass pitch sequences obtained from the sample songs and stored in the file 'bassPitches.dat'.

These four finite state automata independently generate note attribute sequences which are combined to produce a sequence of complete notes. The section above describes the process involved in extracting note attribute sequences from the created finite state automata. This note sequence, spanning two bars, is generated when the element is created and stored in the element. The finite state automata may again be utilised to modify or regenerate the element's note sequence. Modification and regeneration techniques are discussed further in the "Note Sequence Modification Techniques" section.

Percussive elements use a subset of pitches which correspond to instruments included in the instrument type represented by the element. For example, pitch data obtained for the element Hi-Hat Closed may contain values 42, 44 or 80 corresponding to instruments Closed Hi-Hat, Pedal Hi-Hat and Mute Triangle. To keep consistency between songs, percussive elements receive a fixed pitch value that is not retrieved from a finite state automata.

Velocity values retrieved from the source data for the same instrument type, e.g. piano, were found to be inconsistent between songs. Values extracted from a song for one instrument type are dependent on a number of factors specific to that song. One factor is the difference in MIDI instruments that may be assigned to a particular instrument type. For example, the bass instrument type may be played by an 'acoustic bass' in one song and the 'finger bass' in another. These instruments vary in their attack, decay and volume intensity settings. This results in the same velocity producing different volume levels.

Another factor affecting the comparability of velocity values between songs is the associated instrument's volume settings. A low velocity value for the note combined with a high volume setting for the track can produce a similar volume level to a high velocity value and low volume setting. Also, MIDI instrument volume settings are dependent on the other instrument's volume settings, further distorting meaningful comparisons between songs.

To address this, the element object is able to scale the velocities. A fixed minimum and maximum velocity value is assigned to each element. The minimum and maximum values were selected and refined through trial and error. The scaling of the velocities occurs during initialisation and after modification of the note sequence.

3.3.4 Bossa Nova

The instrument set selected for the generated Bossa Nova songs consists of a melody, piano, guitar, bass, strings, bass drum, clave, hi-hat closed, hi-hat open, ride, three different percussive instruments, a piano solo and a percussive break.
The instruments selected were common to the source MIDI files and songs listened to.

The piano instrument behaves in two distinct ways. A piano solo occurs during the middle of the song and the piano provides a sparse accompaniment (comping) during the later phases. The different roles of the piano are treated as two, separate elements. That is, a different object is created for each, using different source data.

The three different percussion instruments are also achieved using three separate elements. Each object is created using the same source data for the pitch, position and duration note attributes. The pitch value is randomly selected for each object. This value determines the percussive instrument assigned to this object. The list of possible instruments consist of a muted high conga, an open high conga and a low conga. These three were chosen due to their widespread use in the Bossa Nova style.

The percussive break element provides a rapid succession of percussive hits leading to a change in the song. The purpose of this element is to improve the dynamics of the song, thus providing a more 'human-like' feel to the song.

3.3.5 Trance

The Trance instrument set overlaps the Bossa Nova instrument set. For example the instrument type 'bass' is common to both musical styles. The instrument set selected for the Trance genre songs consists of an arpeggio, bass, bass drum, hi-hat closed, hi-hat open, melody one and melody two. Also used are, percussion, sweep, a sound effect, build, snare and strings.

The string instrument includes three different elements, a string chord, high and low string. Each element is created from different source data. This separation facilitates the soloing of a high string or a low string common to the Trance genre. The sweep element shares the source data used for the string element but is assigned a different MIDI instrument. Both melody one and melody two also share the same source data. The sharing of source data between similar instruments reduces the amount of source data needed.

3.4 Note Sequence Modification Techniques

A number of different techniques are used to repeatedly transform an element's note sequence resulting in a series of notes spanning the length of the song. The variation achieved by each transformation attempts to imitate the characteristics of instrument voicings found in the genre. For example, the modification of the piano element's note sequence attempts to instil improvisational qualities common to the Bossa Nova style.

The application of these techniques facilitate a more accurate modelling of the genre and avoids monotony, a trait computer generated music is susceptible to. An ending to the song is also achieved in this way. Techniques applied include the partial and complete regeneration, mutation and morphing of note sequences.

3.4.1 Storage and Access

Each element is stored together in one container class. Throughout the generation of the MIDI file, each element responds to external requests, providing their note sequences and altering their state (note sequence) as instructed.

The activity table, a global representation of the song discussed later, holds within its cells the type of technique to be applied. As the cells are traversed, column by column, each of the element's note sequence is retrieved and placed adjacent to its previous one. Also, the technique type is passed as a parameter to the element. This invokes the element's note sequence modification method createVariation().

3.4.2 Bossa Nova

The following instruments from the Bossa Nova style were identified as requiring a dynamic note sequence; melody, both roles of the piano, background and soloing, the percussive build, strings, bass and guitar. The following paragraphs discuss note sequence modification techniques utilised to model the stylistic characteristics displayed by these instruments with in the Bossa Nova style.

The melody element personifies the melody of the song as it would be sung in a live performance. The technique used to achieve a developing melody is regeneration. Each time the note sequence is retrieved a new note sequence is constructed. Position, pitch, duration and velocity values are generated and combined in a way identical to when the melody object was constructed.

The independent chain of note sequences generated contained erratic pitch and velocity intervals resulting in a non flowing melodic tune. To address this issue, a criteria set was defined for a valid, regenerated note sequence. The first criteria determined whether the difference between the pitches of the joining notes was within an acceptable range. The range selected was seven semitones. The velocity scaling, described previously, minimised the discrepancy between joining notes.

The second criteria required - should the last note of the original note sequence commence close to the end of the phase, then the beginning note of the new note sequence must commence at the start of the phase. This was designed to smooth the link between note sequences by ensuring a melodic phrase did not end abruptly. To implement the timely end to a melodic phrase, the note sequence was truncated to a single note. This note was subjected to the same constraints as those specified by a complete regeneration.

Two separate piano elements are used to realise the two different roles a piano performs during a Bossa Nova song. The element modelling the chord-based, background behaviour utilised the regeneration technique described previously. This element provides a sparse, rhythmic accompaniment and therefore no constraint criteria is applied to the regeneration of a new note sequence.

The second piano element models the piano solo. Continual regeneration was used to produce a solo spanning multiple phases. Applying the previously described constraints had the effect of severely limiting the diversity of note sequences generated. For this reason, no constraint criteria was applied to the regeneration of the adjoining note sequence.

Continuous regeneration was also applied to the percussive build element. This avoided a computer generated feel created by identical builds leading to a new section. No constraints were applied to the regeneration of a percussive build element because there are no adjoining builds and different builds do not share any relationship.

The string element provides a new note sequence through the regeneration of its note sequence. The regenerated note sequence does not need to conform to any criteria. The regeneration of the string element achieves a more interesting texture to the music and more closely represents the varying successions of chords found in the Bossa Nova genre.

The bass and guitar elements are regenerated at regular intervals. This provides a better distinction between sections within the song. Data sequences extracted from the source MIDI files do not differ enough to warrant any constraint criteria in the regeneration of the note sequence. The bass and guitar help provide the rhythmic base for the song. For this reason, these instruments need to be reasonably stable and consistent.

The ending of a Bossa Nova song is implemented through the truncation of all element's note sequences. The following elements play a single note at the beginning of the last phase; melody, piano, guitar, bass, bass drum, clave, hi-hat closed, hi-hat open, ride, percussive break. The remaining elements (the three percussion instruments and the string) are silent.

3.4.3 Trance

The following Trance instruments dynamically change their note sequence throughout the creation of a song - arpeggio, melody one and two, string, high string and low string and the percussive build. The techniques used to create variations of the original note sequence are morphing, mutating and regenerating.

The arpeggio element represents the step-based, synthesised note progression common to the Trance genre. The desired effect is one that is bubbly, flowing and gradually evolving. A morphing technique is used to achieve this. The morphing technique is applied when the arpeggio element is active for four consecutive phases.

The morphing technique is applied to the velocity values within the note sequence. A new sequence of velocity values, of equal length to the existing note sequence, is generated. These new values are generated from the finite state automaton which modelled the velocity values obtained from the sample arpeggio sequences.

The acceptance of the new velocity sequence is subject to the following criteria. Firstly, the new values must be different to the original values. A morph between two identical values will result in no change. Secondly, values generated must not be identical to each other. A more interesting effect is achieved when individual notes morph towards different values.

The morphing of the original velocity to the new one spans four phases. Between each of these phases every velocity value undergoes a linear movement towards its destination value. An arpeggio element active for sixteen phases will morph towards a new velocity sequence every four phases.

Table 3.1: An example of morphing by velocity values from the arpeggio element

Phase 1 (original)	Phase 2	Phase 3	Phase 4	Phase 5 (destination)	Phase 6	Phase 7	Phase 8	Phase 9 (destination)
116	102	88	74	**60**	70	80	90	**100**
100	90	80	70	**60**	72	84	96	**108**
104	93	82	71	**60**	72	84	96	**108**
96	87	78	69	**60**	73	86	99	**112**
112	98	84	70	**56**	66	76	86	**96**
108	94	80	66	**52**	65	78	91	**104**

The melody element models a repetitive melodic phrase that occurs throughout a Trance song. This melodic phrase often undergoes a small transformation, occurring on the fourth repetition, before returning to original phrase. To achieve this transformation, a partial regeneration of the melody element is performed.

A new set of pitches are generated for the melody element. Pitches are generated using the same method as when the melody element was created. The current sequence of pitches is saved before the new set of pitches temporarily replaces them.
The rhythmic properties provided by the position, duration and velocity are retained. This provides an association between the original musical phrase and the new, mutated phrase.

The three, separate string elements (strings, high string and low string) realise the behaviour of the synthesised string instrument common to the Trance genre.
The continual regeneration of each string element creates varying quality found within the Trance genre. This variation is bounded by the limited variation found within the source data. The percussive build is also continually regenerated further adding to the dynamics of the song.

Similar to the endings of the songs created in the Bossa Nova style, elements producing the Trance song are truncated to one, or zero, notes. The elements active in the final phase of the song provide a staccato accent at the beginning of the phase.
The elements selected to be active are the bass, bass drum, hi-hat closed, hi-hat open and the percussive build.

3.5 Instilling Global Characteristics

In order to generate a song accurate to a specific genre, the macro characteristics of a song must also be modelled. The implementation of the global structure for a Bossa Nova song is a fixed, expanded activity table, described later. The Bossa Nova style adheres to a strict form in which subtle variations may be achieved through a more simple method. But due to time restrictions this aspect has not been explored. The following sections describe the method used for the implementation of global structure in a song conforming to the Trance style.

An activity table is used to represent the global structure of the song. Each column within the table denotes a phase of the song. Each row models the behaviour of a group of instruments during the song. A cell indicates whether or not the group of instruments is active during this phase.

A genetic algorithm evolves a solution activity table. A genre specific rule set provides a criteria for judging the fitness of each activity table. Instrument groups within the solution activity table are expanded to include all elements involved in the creation of the song. In each cell of the expanded table, the behaviour of the element, within the phase is incorporated with the activity of the element.

3.5.1 Activity table

The activityTable object represents the global structure of a song. The length of an activityTable determines the length, in phases, of the song it represents. A phase is a time length of two bars. Each row represents an instrument group's activity as the song progresses. Each cell contains a Boolean indication of whether the instrument group is present, or not, during the respective phase.

Elements sharing similar behaviour traits have been grouped together in order to achieve a more abstract depiction of a Trance song. The following are the ten instrument groups that have been formed, arpeggio; melody one and two; bass; bass drum, hi-hat closed and hi-hat open; snare; percussion; strings and high string and low string; sound effects and sweep; percussive build.

This grouping of instruments produces an activityTable that is more responsive to the genetic algorithm process. Figure 3.2 is a displays a graphical representation of an instrument activity table.

On construction of an activityTable, Boolean cells are set according to a rule sequence. The rule sequence instils a macro pattern that is common to the Trance genre. The rules are not violated by the genetic algorithm and remain intact in the solution activityTable.

arpeggio	*	*	*	*	/	/	/	*	*	...	/
melody	/	/	/	/	*	*	*	*	/	...	/
bass	/	/	/	/	*	*	*	*	/	...	/
bass drum	/	/	*	*	*	*	*	*	/	...	/
hi-hats	/	*	*	*	/	/	*	*	*	...	*
snare	/	*	*	*	/	/	*	*	*	...	*
percussion	*	*	*	*	/	/	/	*	*	...	/
strings	/	/	/	/	*	*	*	*	/	...	/
sound effects	/	/	/	/	*	*	*	*	/	...	/
build	/	/	*	*	*	*	*	*	/	...	/

* - active
/ - inactive

Table 3.2 Instrument activity table

3.5.2 Rule Sequence

Each column, or phase, within the activityTable has a number of instrument groups that are set to active for that phase. This number of active instrument groups defines the size of a phase. The rule sequence produces a pattern of phase sizes. This pattern permits a large degree of variation, through randomness, while maintaining a degree of instrument group organisation familiar to Trance songs.

The sequence of rules are applied to each phase sequentially. The size of the phase is determined by its position within the song and the size of the previous phase. The following are the rules which shape the pattern; no phase size increment is greater than two and no size decrement is greater than one, a drop occurs when both phase position is divisible by four and phase size is greater than six, a phase size can not be zero or greater than the number of instrument groups.

To implement these rules, without severely limiting the variation of dynamics possible, the following logic was used.

```
// assign a random phase size, for the first phase, between 1 and 3
phaseArray[0].size := random(1,3);
for each phase in theSong                    // iterate through each phase in turn
        // if the phase is at the beginning of a 4 phase segment
        if (phase.position MOD 4 = 0)
                if (phase.size >= 7)
                        // generate a number between 1 and 3
                        dropSize := random(1,3);
                        if (dropSize = 1) then              // reduce by a third
                                phase.size := phase.size - (phase.size/3);
                        else if (dropSize = 2) then         // reduce by 2 thirds
                                phase.size := phase.size - (phase.size/1.5);
                        else if (dropSize = 3) then         // reduce to 1
                                phase.size := 1;
                        end if;
                else                                        // increase or maintain size
                        phase.size := phase.size + random(0,2);
                end if;
        else                                                // increment or decrement
                phase.size = phase.size + random(0,2) - random(0,1);
        end if;
        if (phase.size > numInstruments) then               // check boundaries
                phase.size := numInstruments;
        else if (phase.size < 1) then
                phase.size := 1;
        end if;
end for;
```

Figure 3.13: Pseudo code illustrating the implementation of the rules

Once the size of each phase has been determined, instrument groups are randomly activated to satisfy this size. This is done in the following way. Instrument group settings from the previous phase are copied to the next phase. The difference in size is then calculated, instrument groups are then randomly activated, or deactivated, as required.

For example, a phase with size seven following a phase with size six will contain the instrument groups active in the previous phase. In addition, this phase will have one more instrument group active. This instrument group is randomly selected from the remaining, inactive instrument groups. Instrument groups required to satisfy the size of the first phase are all randomly activated.

3.5.3 Genetic Algorithm

A genetic algorithm is a search strategy that provides solutions to complex problems. The genetic algorithm models the evolutionary process. "They [genetic algorithms] view learning as a competition among a population of evolving candidate problem solutions."(Luger, G., 2002).

A population is formed by encoded representations of potential solutions. This population evolves through the application of evolutionary operators - crossover and mutation. Individuals of the population are evaluated by a critic and assigned a fitness. Individuals receiving a high fitness value are favoured in survival and reproduction. Ultimately, an individual is translated back into the problem domain as a possible solution.

A population of activityTable objects possess the behaviour needed to interact with the process of a genetic algorithm. An activityTable object is initialised to a valid state on construction. An activityTable object is able to evaluate its own fitness and can also perform a valid mutation when requested.

The genetic algorithm process used in this study involved the creation of a population of forty activityTable objects, each object of a fixed length. 500 generations were then performed. Within each of these generations the population determined their individual fitness, were ranked according to this fitness and were subjected to crossover and mutation operators. The following paragraphs describe this cycle in detail.

Fitness Evaluation

A rule-based approach is used in the evaluation of an activityTable's fitness. Each activityTable object is subjected to the same criteria. The implementation of the evaluation is in the form of functions. A function returns a numeric penalty, or bonus, based on the arrangement of activity within the table. Functions examine the activity table for instrument activity patterns common to the Trance genre. The following paragraphs describe the instrument group behaviours and relationships tested.

A Trance song may be logically partitioned into segments of four phases. The behaviour of certain instrument groups are examined for conformance to this partitioning. A sequence of active values for the instrument group must comprise of a sequence entry point beginning at the start of a segment and an exit point at the end of a segment. A further penalty exists for sequences of active values that are of a length less than four or not divisible by four.

This criteria is applied to instrument groups which exhibit a consistent, flowing ambience in Trance music. They are the arpeggio, both melodies, the bass, all percussive instruments and the strings. A penalty of one is given for each violation of the constraints. The aggregated penalty assigned from the evaluation of all instrument groups is weighted via the doubling of the value. This weighted value is then tested against a range constraint. If it exceeds this value it is again doubled.

To achieve the dynamics of a Trance song, a range constraint exists specifying the total number of phases an instrument group may be active. This ensures the resulting table is not too sparsely populated or too busy. The instrument groups subjected to this constraint are the strings and the instrument groups comprising the beat of the song. That is, bass drum, hi-hats and snare. Each are assigned different minimum and maximum values depending on the frequency required of them.

Instrument groups that do not conform to this constraint are awarded a penalty of four. If the number of active elements is four more than the maximum, or four less than the minimum, a penalty of eight is awarded. The aggregate penalty is checked against the maximum allowed value and is doubled if it exceeds this value.

Resulting songs lacked the driving drum beat characteristic of a Trance song. To amend this, a bonus of negative one was awarded for each phase containing the following active instrument groups, bass, bass drum, hi-hats and snare.

A reduction in the phase size by a value larger than one is referred to as a drop. Drops occur at the beginning of a segment. They provide a quick release to a slow build up and help create a distinction between sections within a song. The number of drops and the nature of the instrument groups during a drop are assessed.

The desired number of drops within a generated Trance song, thirty two phases in length, is between three and five. This achieves a balance between the building and releasing of tension. A penalty of fifteen is assigned to an activityTable comprising of two or six drops. Less than two drops, or more than six, attracts a penalty of twenty five.

A drop usually precipitates a change in the music. This change can be achieved by the activation of one of the primary, melodic instrument groups that was not present in the previous phase. If the arpeggio, melody one and two, strings is active a bonus of negative one is awarded. If only one of these instrument groups is active, and this instrument group was not active in the previous phase, a bonus of negative three is given.

The following constraints provide a dependency between instrument groups. Inter-instrument relationship flaws, in the context of the Trance genre, were identified in generated song. These flaws are addressed via the penalising of the fitness in the following ways.

A phase with only percussive instrument groups active receives a penalty of one. The simultaneous activation of the arpeggio and melody one or two instrument group results in a penalty of one. The occurrence of a phase containing the bass drum and snare instrument group without the hi-hat instrument group attracts a penalty of one. The occurrence of a phase containing the snare instrument group without the bass drum or hi-hat instrument group also attracts a penalty of one.

The fitness value assigned to the activityTable is the aggregate of all penalties and bonuses awarded. The activityTable performs the tests described above and assigns itself the resulting fitness value. This value is then used to sort the population and select the members that are kept and those which are deleted.

A negative score is often achieved by individuals subjected to the genetic algorithmic process. This score was obtained by the best individual after the 500 generations. An analysis of the composite penalties and bonuses involved in the better of these scores indicate that all constraints were successfully satisfied.

Crossover

An elitist strategy is employed. The top two fifths of the population are retained, as is, and the bottom three fifths are deleted. The remaining members are potential parents which replace the deleted members with their children. Parents are randomly selected and the children are produced using two point vertical crossover or two point horizontal crossover.

Providing both selected parents are not the same, two children are produced using two point crossover. A vertical crossover is four times more likely then a horizontal crossover. The combination of horizontal and vertical crossover techniques increased the efficiency of the genetic algorithm in achieving complex, instrument group patterns.

A two point, vertical crossover involves two parent activityTables and two child activityTables. Two random points are generated between zero and thirty one (an activityTable is thirty two phases in length). A check is performed determining whether

a crossover using these points will violate the sequence rules used in constructing the activityTable. A set of new points is regenerated until a valid crossover can take place.

A crossover point check helps provide a seamless integration of parent phases. Phases being joined must adhere to the sequence rules, an increment greater than two or a decrement greater than one results in a fail. Also, should the instrument groups bass, bass drum, hi-hats, snare sequence be active in the previous phase, they must be active in the phase being joined.

Two point, vertical crossover between two parent activityTables produces children using the following method. Child one receives the following phases; phases zero to point one of parent one, phases from point one to point two from parent two, phases from point two to the last phase from parent one. Child two receives the inverse of these phases, that is; phases zero to point one of parent two, phases from point one to point two from parent one, phases from point two to the last phase from parent two.

Two point, horizontal crossover follows a similar process. Two, valid crossover points are randomly chosen. Rows between these points are copied from the parents to the children. This is illustrated in Figure 3.14.

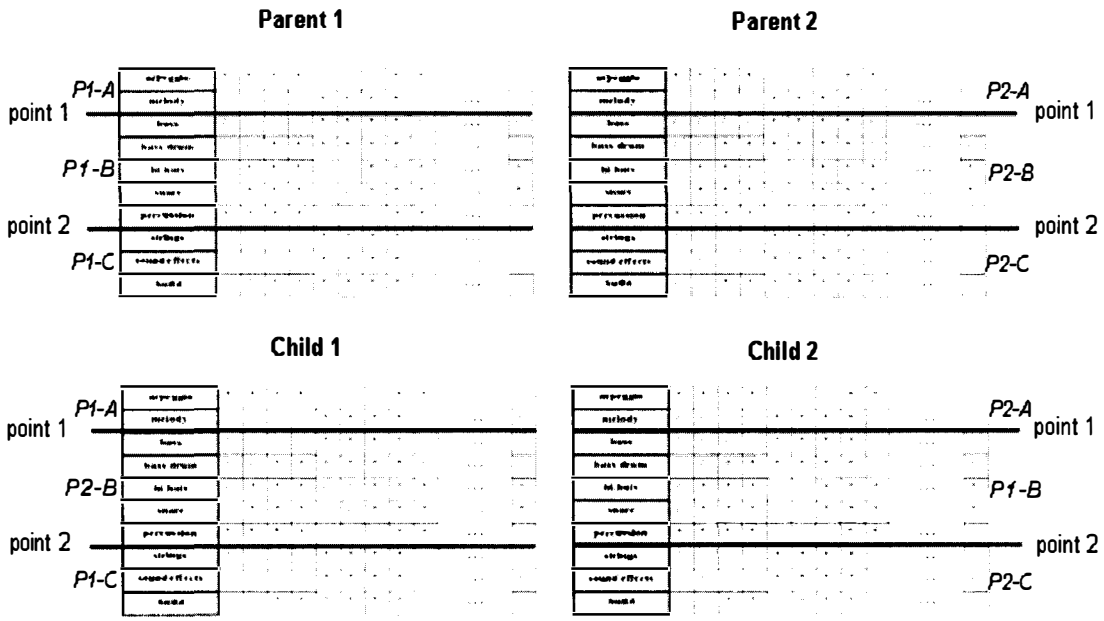

Figure 3.14 Horizontal crossover

Child one receives; rows zero to point one from parent one, rows from point one to point two from parent two, rows from point two to the end from parent one. Child two receives; rows zero to point one from parent two, rows from point one to point two

from parent one, rows from point two to the end from parent two. Vertical two point crossover is achieved in a similar way.

Mutation

Once all deleted members of the population have been replaced, mutation is applied to selected members. The top one fifth of the population is untouched by mutation. Mutation involves the switching of the status of one instrument group within a phase. There is a one in twenty chance that each phase, within a selected activityTable, may be affected by mutation. For example, if a phase is selected to be mutated and an active element group is randomly chosen, its status will be switched to inactive.

Mutation is also constrained by the sequence rules implemented in the construction of an activityTable object. Decrement and increment differentials between phases are limited to one and two respectively. Also, a phase can not have a size of zero. Further, the mutation is restricted from effecting a bass, bass drum, hi-hats or snare instrument group that was active in the previous phase. This aids the preservation of a driving, consistent beat crucial to the Trance genre.

3.5.4 Expanding the Solution

The product of the genetic algorithm is an activityTable adhering to the sequence rules instilled at initialisation and maintained throughout the process. The activityTable also displays characteristics formed by the process and guided by the evaluation criteria specified. The solution activityTable selected is the member of the final generation/cycle with the highest fitness.

This solution activityTable must undergo an expansion process to encompass all element activity. Instrument groups need to be expanded to specify the activity of each of the elements they represent. Note sequence variation techniques each element must undergo are also set for each instrument during this stage.

Rows of the activityTable belonging to an instrument group representing a single element are linked to this element. The melody, hi-hats, strings and sound effect instrument groups are expanded and set depending on neighbouring information. The following paragraphs describe the rules behind each instrument group expansion.

The melody instrument group represents both the melody one and melody two elements. An A|B|A form is incorporated into the resulting song. This is achieved

through restricting the appearance of melody one to the A sections and melody two to the B section.

The hi-hats instrument group consists of the hi-hat open and hi-hat closed elements. An expansion of this group will result in one, or both, being active. Which element(s) is active is dependent on the size of the phase, the position, within a segment, of a phase, whether a drop has occurred and which elements were active in the previous phase.

A lookup table implements the rules for expanding an active hi-hats group. These rules are displayed in the table below.

Table 3.3: Expansion rules for the hi-hats instrument group

hhc previously on	hho previously on	drop	aligned with segment	big	result
yes	yes	yes			hhc only
yes	yes	no			both
yes	no		no		hhc only
yes	no		yes	yes	both
yes	no		yes	no	hhc only
no	yes		no		hho only
no	yes		yes	yes	both
no	yes		yes	no	hho only
no	no			yes	both
no	no			no	random

In the above table 'hhc' refers to hi-hat closed and 'hho' refers to hi-hat open. The columns 'hhc previously on' and 'hho previously on' indicate whether the respective hi-hat type was active in the previous phase. A phase containing more than five active instrument groups is classed as 'big'. These rules were derived from listening to Trance style songs. The two, key characteristics these rules attempt to instil are consistency and a responsiveness to the dynamics of the song.

The strings instrument group contains the elements strings, high string and low string. The selection of an active element(s) is dependent on the following features. Whether the element was active in the previous phase, whether the current phase is

during the beginning or ending of a song, the size of the phase, the position of the phase and whether the bass element is active.

These rules are implemented through a lookup table similar to the one used in expanding the hi-hats instrument group. The table below shows the rules used in expanding the string instrument group.

Table 3.4: Expansion rules for the string instrument group

stringHi previously on	stringLo previously on	strings previously on	intro or end	big	aligned with segment	bass currently on	result
no	no	no	yes			yes	stringHi
no	no	no	yes			no	stringLo
no	no	no	no	yes			strings + stringHi
no	no	no	no	no			strings
no	no	yes			no		strings
no	no	yes		yes	yes		strings + stringHi
no	no	yes		no	yes		strings
yes	no	no			no		stringHi
yes	no	no			yes		strings + stringHi
no	yes	no			no		stringLo
no	yes	no			yes		strings
yes	no	yes			no		strings + stringHi
yes	no	yes		yes			strings + stringHi
yes	no	yes		no			strings

Within the above strings instrument group lookup table, 'stringHi' refers to the high string element and 'stringLo' refers to the low string element. The first three columns indicate whether the element denoted was active in the previous phase. A phase is classed as 'big' if it contains more than five active instrument groups.

The aim of the lookup table is to provide consistency between phases. A phase that is not aligned with a segment will include the element(s) active in the previous phase. Also, the size of a phase will dictate the number of string elements activated.

This will prevent the members of the strings instrument group from overwhelming, or being overwhelmed by, the other elements.

The final instrument group to be expanded is the sound effects group. This group consists of two members, the sound effects element and the sweep element. The selection of which element(s) to activate is based on whether the element was active in the previous phase, the position of the current phase and whether there is a drop occurring in this or the next phase. The following table displays the rules used to determine the active element(s). This table refers to the sound effect element as 'sfx'.

Table 3.5: Expansion rules for the sound effects instrument group

sfx previously on	sweep previously on	next phase is a drop	current phase is a drop	aligned with segment	result
		yes			sweep
			yes		sfx
				no	sweep
yes	no			yes	both
no	yes			yes	both
yes	yes			yes	both
no	no				sfx

The following is the general element behaviour achieved by these rules. A sound effect will occur at the start of a segment, this is because a drop will always occur at the beginning of a segment. A sound effect is more common than a sweep. A sweep will occur approaching a change, drop, in the music.

The expansion of an activity table also involves the setting of the type of note modification technique to be applied. This is done for each element in each of the phases of the song. The types of note modifications and when they are triggered are explained in the previous section 'Note Modification Techniques'.

3.6 MIDI File Rendering

From the previous processes an internal representation of the song has been created. This is in the form of an expanded instrument activity table. The table holds the

global form of the song and orchestrates the integration of the elements required to produce the local aspects. This table is a template, guiding the generation of the song in a way similar to that of a class defining the behaviour of its instance.

The generation of a song is the conversion of the internal representation into the machine-readable format MIDI. This allows the auditioning of the song by software that is able to translate the MIDI format into music. The conversion to MIDI also provides portability between different computers and OS platforms. This portability assists in obtaining an objective evaluation, discussed later.

3.6.1 Chord sequence

A fixed chord sequence is set for songs generated in the Bossa Nova and Trance genre. The chord sequence used in generating each Bossa Nova song is taken from the song 'How Insensitive' by Antonio Carlos Jobim. The fixed chord sequence applied in the generation of trance songs is a repetitive, minor sequence typical of the Trance genre.

Within the Bossa Nova genre it is common that multiple chord changes will occur during one phase (two bars). For this reason, the chord sequence specifies two chords for each phase in the song, one for the first bar and one for the second.
The Bossa Nova song, from which the chord sequence was emulated, contains no more than one chord per bar.

The internal representation of each chord in the sequence contains the following attributes necessary to transpose the note sequence. The amount each note must be lowered or raised, the tonality (only major or minor is implemented), and the chord's voicing. Notes are referred to as their position within the twelve note octave.

The note sequences held, by each element, are generated from the source data obtained from the sample songs. This data was lowered or raised to the key of C whilst maintaining the tonality of the note sequence. The implementation of the chord sequence requires each note in the note sequence be transposed to the designated chord.

Modifying the pitches of the note sequence to conform to the designated chord require two processes. The generation of the 'valid' notes derived from the designated key and the decision of how 'invalid' notes are amended. Also, the increment or decrement of note pitches to the designated key. The latter process is implemented through the addition of the chord representation's first attribute to each note in the note sequence.

To address the first process, a position-centric representation of both the major and minor scale is stored. This defines the positions, within one octave, in which valid notes reside. The treatment of the altering or extending of chords, such as an added seventh or flattened fifth, is illustrated in Figure 3.15. The remaining notes that reside outside of the scale, defined by the chord, are raised to the next, valid note.

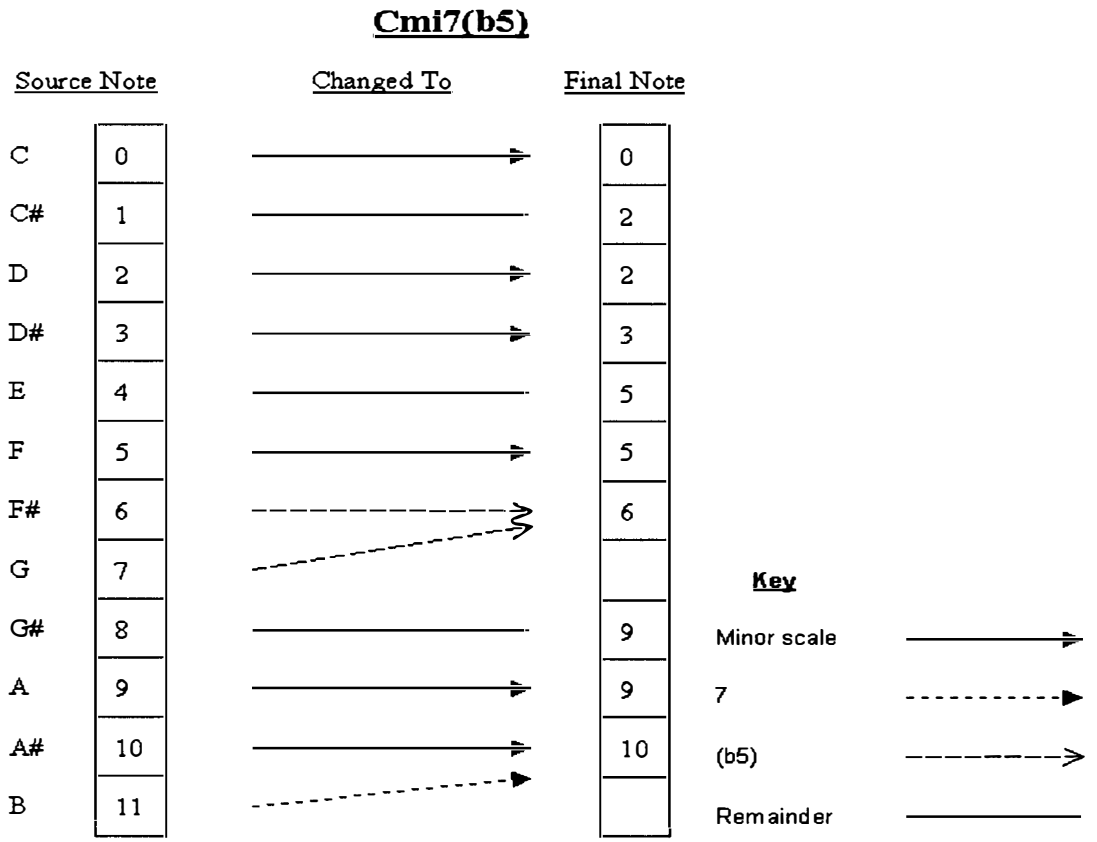

Figure 3.15: Depiction of the application of a chord to a note

3.6.2 Note Sequence to MIDI Conversion

The internal representation of the note sequence, contained within each element, must be converted into a MIDI format and placed in the correct position within the appropriate track. The following steps achieve this conversion. Creation of a corresponding MIDI note-on and note-off event for each note in the sequence. Sorting the array according to note event times. Check for notes exceeding the two bar limit. Inserting the events into the appropriate track

Firstly, for each note in the sequence, a note-on and note-off event is created. As described before a MIDI note event consists of a status, time, datal variable (pitch), data2 variable (velocity) and a data3 variable which is not used by a note-event. The generation of a note-on event for a note is achieved through the following process.

The status value is set to 144, indicating a note-on event. The time is set to the position value of the note. The data1 variable is set to the note's pitch value and the data2 variable is set to the note's velocity value. Next, the generation of a note-off event occurs for the same note. This involves the setting of the following values to the note-off event.

The common method of indicating a note-off event to the MIDI sequencing application is through the use of a note-on event with a velocity of zero. For this reason, the status of the note-off event is also set to note-on. The time is assigned the value of the note's position plus the note's duration. These two values combined provide the position, in ticks, of where the note ends.

The data1 variable is set to the current note's pitch. This provides a link between the note-on and note-off events implying that both events belong to the same note. The data2 variable is set to zero to denote a note-off event. These events form a sequence that may not be in the correct order.

For example, the order of events generated from the occurrence of a note ending after the beginning of the subsequent note is incorrect. The note-off event for the first note will be positioned behind the note-on event for the second note despite occurring at a later time. To remedy this, the array of note events is sorted based on their assigned time values.

The next step is ensuring the last note does not exceed the two bar limit of 960 ticks. If it does, the note's time value is truncated to 960 ticks. The next stage of the process is inserting the MIDI events into the appropriate track. To do this, the accumulative time values assigned to the MIDI events must be changed to a relative value and then stored according to the element they correspond to.

A relative time value indicates, in ticks, the time since the last event occurred. This value is calculated by keeping a running total of the time elapsed and subtracting this from the value originally assigned. The modified MIDI event is then stored in the track assigned to the element.

Part of the CMaxMIDI toolkit, provided by Paul Messick, is used to create and manage MIDI files and MIDI tracks. Two classes, containing calls to window's dll files and library functions, facilitate the creation of a blank MIDI file, the generation of empty tracks and the setting of MIDI file parameters. The toolkit is supplied with the purchase of the book "Maximum MIDI : Music Applications in C++" by Paul Messick.

3.6.3 Instrument Activity Table Traversal

A blank MIDI file is created. The resolution of the MIDI file is set to 120 ticks per quarter note. The tempo of the song is specified. A number of empty tracks are then created; one for each element involved in the creation of the song, one containing global settings of the MIDI file.

Tracks assigned an element are designated a general MIDI (GM) instrument. The GM instrument is selected from a list of 128 instruments. This list is standard across sound cards, computers and manufacturers. To assign an instrument to a track, MIDI events are created which specify the channel, the bank containing the instrument and the position of the instrument within this bank. The instrument assigned to the track performs the notes provided by the element designated to the track.

The name of the instrument type represented by the element is assigned each track. This name is visible when viewing the MIDI file through a MIDI sequencer application. An initial time offset is stored for each track. This offset value takes into account the MIDI events stored during the setting of the instruments and ensure the first note of each track occurs simultaneously.

Once the tracks have been set and the MIDI file initialised, the supplied, expanded instrument activity table is traversed. A column by column traversal elicits the information required to form the desired note progression for each track. These resulting tracks are attached to the MIDI file. This file is then saved and closed.

The pseudo code on the following page demonstrates the process of extracting the required information from the table.

```
// iterate through each phase in the song
for each phase in song.length
        // iterate through each element in the phase
        for each elem in phase
                // retrieve chords for each bar of the phase
                phaseChords[] = song.getChords( phase );
                if ( elem = Null ) then              // if element is not active in this phase
                        // increment offset with two bars of rest, (960 ticks)
                        offsetTime[elem.position] = offsetTime[elem.position] + 960;
                else    // create a copy of the element
                        tempElem := new element(elem);
                        // if the element is not percussive, transpose the note sequence
                        if ( NOT tempElem.isPercussive() ) then
                                tempElem.applyChords( phaseChords[] );
                        end if;
                        // convert the note sequence into MIDI events and store in track
                          designated to elem
                        processElem( elem, elem.position, offsetTime[elem.position]
                        offsetTime[elem.positon] := 0;    // reset offsetTime for element
                        delete tempElem;                  // delete copy of element
                end if;
                // retrieve the note modification technique
                char type = phase.getVariant(elem.position);
                elem.createVariation( type );             // apply note modification technique
        end for;
end for;
```

Figure 3.16 Pseudo code demonstrating the traversal of the instrument activity table

3.6.4 Bossa Nova

The implementation of the Bossa Nova, song conversion process differs to the method used to generate a Trance song. The different set of elements contributing in the generation of a Bossa Nova song requires the assignment of different GM instruments. An additional structure is employed in the generation of melody element's note sequence.

The GM instruments assigned to each track were selected based on the following; their accuracy to their counterpart instrument in live recordings of Bossa Nova songs, the GM instruments selected in the sample MIDI files, and the sounds provided by the hardware dependent soundcard used to audition the songs.

The melody and voice are performed by the 'Shakhchi' instrument. This instrument elicits a breathy, flute sound emulating the breathy voice of a Bossa Nova singer. The 'Grand Piano' is selected to play the piano solo and background accompaniment. A 'Nylon Stringed Guitar' performs the notes generated by the guitar element. The 'Synthesized String 1' instrument best emulated the string element and the 'Acoustic Bass' was assigned to the bass track. A 'Jazz Kit' provided the percussive instruments.

Within the form of a Bossa Nova song, a structure to the melody is present. One, melodic note sequence may be interpreted as a string of musical phrases. The repetition of a rhythmic sequence or the transposed. Partial repetition of a musical motif may form a association between different musical phrases. To emulate this, the following structure is imposed on the generation of the melody track.

The melody element is active for sixteen phases before and after the piano solo. The notes generated for the first sixteen, consecutive phases are stored. These notes are then retrieved to form an identical note sequence for the second, sixteen phases. Further, within these sixteen phases a pattern is imposed using the following steps.

The melody element provides four different note sequences for the first four phases. These four phases are then repeated, in the same order, providing the following four phases. The application of the chords designated to each phase effects note sequences produced by the melody element. This results in latter, four phases differing to the former yet still bearing a resemblance.

Phases nine and ten receive new, regenerated note sequences from the melody element. These two, new note sequences are repeated twice. The final two phases of the sixteen receive new, regenerated note sequences from the melody element. The following diagram graphically represents the form of the sixteen phases where each section has a length of two phases.

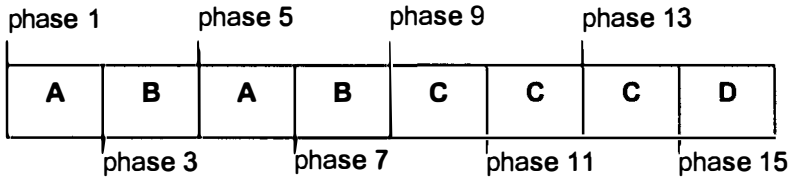

Figure 3.17 Graphical representation of the melody structure

Note sequences preceding phase nine and phase sixteen are truncated. The effect of this is the segregation of segments of the sixteen phase form. Musical phrases are defined within this form by this segregation and the repeated rhythmic and melodic aspects.

3.6.5 Trance

Songs generated in the Trance style follow the steps outlined in the 'process' section without the variation described above. The GM instruments assigned to perform the notes generated by each element were more difficult to select. The instrument set provided by the general MIDI standard lacks the synthesised instruments common to the Trance genre.

The following instruments selected most closely emulated the desired sound. A 'Muted Electric Guitar' was assigned to reproduce the synthesised, arpeggio effect required by the arpeggio element. The 'Synthesised Bass' was assigned to the bass track. A 'Standard Kit' provided the percussive, Trance instruments.

The 'Lead 8 (bass + lead)' instrument was assigned to the first melody track and the 'FX 1 (rain)' was assigned to the second melody track. 'FX 6 (goblins)' performs the notes produced by the sweep element. A 'Reverse Cymbal' performs the sound effect element. The tracks containing the notes for the high string, low string and strings are all assigned the instrument 'Pad 7 (halo)'.

3.7 Obtaining an Objective Critique

To achieve the evaluation of resulting compositions, a survey was designed and conducted. This survey aimed to generate objective feedback which answered the research questions posed, assessing the technical and aesthetical quality achieved.The survey also provided a measure of the conformance to the intended genre, established whether the song's non-human origin was evident, and graded the overall quality of the songs in terms of usefulness.

Obtaining an accurate, objective measure of a subject that is inherently subjective is difficult. To help address this issue, a small number of songs were generated for assessment. This allowed a greater number of different evaluations for each song. By analysing multiple evaluations, a more accurate measure of the song's performance may be gauged from the common trends that emerge.

Participants were selected from students attending a Jazz course in the Western Australia Academy of Performing Arts (WAAPA). Students selected were accustomed to providing 'real-time' feedback for a variety of genres and possessed a sound knowledge of these genres acquired from their course. The survey was designed to elicit an instinctive response, to each question, by the student. A lengthy, critical examination of each songs by the participants did not seem feasible.

3.7.1 Questionnaire design

The survey consists of two different questionnaires - one used in evaluating each song auditioned and one completed at the end of the session. The first questionnaire facilitates a song-specific evaluation. The second questionnaire is designed to assess the overall quality of all songs evaluated by the participant. Each participant completes multiple copies of the first questionnaire and only one copy of the second questionnaire.

Both questionnaires were designed with consideration to the musical background of the participants. The technologically-based research questions of this project were translated into corresponding questions constructed with musical terminology.
The resulting questionnaires were inspected by a WAAPA lecturer for musical validity. The following paragraphs discuss the intent behind each of the questions formed.

Song Questionnaire

The questionnaire entitled 'Questions for each Song' is attached as APPENDIX B. The purpose of this questionnaire is to obtain technical and aesthetical feedback, from the student, regarding a particular song. The questionnaire consists of one, landscape page containing eight questions. Confinement of the questionnaire to one page simplifies the completion of this questionnaire.

The first, seven questions are in a closed format and require the participant to respond to a *Likert* scale. The optional responses for the participant are Strongly agree, Agree, Neutral, Disagree. Strongly disagree and Unsure. Employing the use of a *Likert* scale facilitates the quick, instinctive response desired. Question eight is an open-question providing the participant with the option to express themselves more freely.

Question one states "*This song is recognisable as belonging to the musical genre.*". This general question allows the participant to consider the song, as a whole, before focusing at a more detailed level. This question encompasses all three research questions. The feedback from this question is important in indicating the success of one of the principle project objectives, composing music that is recognisable to the selected genre.

Questions two to four prompt a more detailed analysis of how the song achieves, or fails to achieve, conformance to the genre. Meaningful, musically valid questions could not be derived that address each research question individually. Instead, questions two to four assess musical aspects of the piece from which answers to the research questions can be derived.

Question two is split into three parts, each part addresses a different aspects of conformance to the genre. Part one is designed to elicit feedback gauging the accuracy of chord voicings with regard to the genre being modelled. The rhythmic properties of the piece is the focus of part two. Part three addresses the musical phrasing.

The aggregated results from these three parts provide a indication of how effectively elements supply and modify their note sequences. This relates to the first and second research questions; "*Can sequences of notes, generated by finite state automata, reflect the characteristics found in the sample data used to create the finite state automata?*", "*Can sequences of notes be affected through morphing, mutation, regeneration, crossover, or a combination of these methods, to accurately represent variations in note sequences typical to the genre selected?*".

Individually, each part of question two relates to a corresponding process within the software. Simultaneous notes, created by elements such as guitar and strings, form the chord voicings within the piece. The rhythmic property of the song comprises of the position and duration values assigned to notes within each element. The arrangement element note sequences form the musical phrases subjected to evaluation by question two, part three.

Question three addresses the conformity to the genre from the perspective of the instruments. The question assess the following aspects; the fulfilment of the role required from the instrument by the genre, the interaction between instruments reproducing interactions typical to the genre. The feedback resulting from the former part of the question is intended to provide an insight into how well the elements model their respective instruments.

The latter part of the question examines a characteristic of the music that is only explicitly implemented at a phase level. Interaction between instruments, within a two bar scale, has not been developed. The intention of this question is to ascertain whether genre-specific, inter-phase interaction, between elements, is achieved through independent combination of each element's note sequence. Put another way, the question tests the assumption that a phase length of two bars provides enough precision to accurately model instrument activity within the desired genre.

Question four, "The form of this song displays a realistic style.", addresses the research question "Can a skeletal structure, representing typical instrument activity in a musical genre, be evolved using a genetic algorithm?". The resulting form of the song is primarily dependent on the instrument activity evolved.

An issue arising early in the development of the project was the generation of stilting, halting, compositions. This effect was largely due to the phase segmentation, quantisizing of position values and problems associated with translating into MIDI. Certain measures were taken to address this issue. Question five assesses the success of these measures and establishes how evident the problem is.

The purpose behind question six was to determine whether the song's non-human origin is evident. Due to the computer-based technology used to audition the music, the question asked the participant to compare this song to other, human-composed songs in the same format.

Question seven was designed to obtain an aesthetical rating, from the participant, for the song being evaluated. This subjective value was expected to be dependent on the quality of equipment and the auditioning method used. The open form of question eight provides the participant with the option to clarify answers they have given and express opinions not addressed by the survey.

Overall Evaluation Questionnaire

The second questionnaire is entitled 'Final Evaluation' and is attached as APPENDIX C. This questionnaire is completed by the participant after he/she has evaluated all songs. The questionnaire contains four questions.

As with the previous questionnaire, this questionnaire is designed to be simple and quick to complete. The four questions are presented, portrait, on one sheet of paper. The first two questions require the participant to tick the box indicating their answer. Question three is an extension to question two and question four is an open-ended question enabling the participant to express comments.

The first question asks the participant to classify the overall quality of the songs auditioned. Possible classifications provided were Novice/Beginner, Amateur, Professional, Commercial Quality and Other. An assumption was made that the distinction between a novice, or beginner, and an amateur was common among participants.

The second question required the participant to form an opinion of the software based on the songs auditioned. Acknowledging that a weak relationship exists between the software's output and the software's functionality, a brief description of the functionality of the software was provided before commencement of the survey.

Also, example applications of the software were listed to better illustrate the meaning of the question.

Question three enabled the participant to elaborate on how they would use the software. The purpose behind this question was twofold; to verify the participant's interpretation of the functionality, and limits, of the software, and to obtain feedback indicating the strengths of the software. The final, open-ended question prompted the participant for further comments. This question enables the participant to better express any general opinions or comments regarding the quality of the songs.

3.7.2 Song Generation and Preparation

The MIDI files comprising the song list auditioned in the survey were generated from the software beforehand and tested. The first, consecutively generated, seven songs from each genre were allocated to the song list. To ensure data from both genres was collected, the order of the fourteen songs alternated between the Bossa Nova and Trance genre e.g. BOSSA01, TRANCE01, BOSSA02, TRANCE02 etc.

The appropriate MIDI-specific settings were assigned to each of the generated files. The song list was then tested on the laptop selected to audition the songs during the survey. Volume settings were adjusted to accommodate; sample sounds assigned to MIDI instruments by the laptop's soundcard, frequency differences in the intended desktop speakers.

3.7.3 Survey Procedure

The survey was conducted with second year WAAPA students enrolled in the course Bachelor of Music (Performance) - *Jazz*. These students were recruited, with permission from the WAAPA lecturer, from jazz lectures which they attend.
Students received an invite to participate in the survey in the form of a short speech at the beginning of the lecture. Following the lecture, students agreeing to participate were lead to a previously booked room.

Each participant was supplied with a booklet containing a disclosure form, 'Questions for each Song' questionnaires and a 'Final Evaluation' questionnaire.
To facilitate an anonymous survey, the booklets were assigned a unique identification. Participants were not required to provide any personal information. Pencils and pens were provided.

The following technical equipment was used to audition the songs; a Pentium laptop PC, a pair of desktop speakers, a number of incentives promised in the invitation. Songs to be auditioned were previously generated and prepared. Students received a brief introduction and a description of the software's functionality. The survey's procedure was then described before the commencement of the survey.

Conducting the survey involved the following steps. The name and style of the song, about to be auditioned, was conveyed to the participants. The participants were required to write this song identification on the appropriate place in the 'Questions for each Song' sheet. The song was then played in its entirety through the desktop speakers. Participants were required to complete their corresponding 'Questions for each Song' questionnaire during this time.

Once all songs in the song list had been auditioned participants were required to complete the 'Final Evaluation' questionnaire. Students unable to remain for the entire duration of the survey were asked to complete the 'Final Evaluation' questionnaire before leaving. Concluding the survey involved thanking the participants, asking for any additional informal feedback and recollecting the distributed booklets.

4.0 - RESULTS

4.1 Introduction

Fifteen students participated in the survey providing feedback for an average of 8.8 songs each. This resulted in between 118 and 131 responses to each question in the 'song evaluation' questionnaire. Also, 15 responses to each question in the 'overall evaluation' questionnaire. The amount of data gathered is analysed to provide an indication of the degree to which the research objectives have been achieved.

Participants completing the 'song evaluation' questionnaire were required to respond to a statement by selecting the desired option from a Likert scale. Values provided for the participants were; 'Strongly agree', 'Agree', 'Neutral', 'Disagree' and 'Strongly Disagree'. The values assigned to each value is 5, 4, 3, 2, 1 respectively with value 99 being assigned to 'Unsure' answers. The mean response provided is based on these numbers.

The following sections individually address the research objectives, research questions, and any interesting, or unexpected, results.

4.2 Automated Composition of Songs in Two Contrasting Genres

The primary objective of this project is the automated composition of music that is recognisable to the selected genre, not obviously "machine generated", aesthetically pleasing and usable in a commercial context. Each of these criterion are addressed separately by questions within the 'song evaluation' and 'overall evaluation' questionnaire.

Within the 'overall evaluation' questionnaire, the participant is asked to classify the quality of songs auditioned as either 'Novice', 'Amateur', 'Professional' or 'Commercial'. A number of participants assigned a separate mark for each of the genres. The need to distinguish between the quality of Bossa Nova songs and the quality of the Trance songs indicates a different degree of success accomplished in each genre.

Table 4.1: Quality classifications designated to the generated songs

Genre	N*	Classification** (per cent)				Mean
		Novice	*Amateur*	*Prof*	*Comm*	
Both	12	27	27	27	0	3.00
Bossa Nova	3	7	13	0	0	3.33
Trance	3	0	7	7	7	3.02

* Number of evaluations
** Prof=Professional; Comm=Commercial

A shown in the Table 4.2, three of the fifteen participants specified a separate rating for each genre. Each of these participants ranked the quality of Trance above the quality of Bossa Nova songs produced. Of the participants who supplied one answer for both genres, none felt the songs produced were of a 'Commercial' quality.

For each song auditioned, participants were required to respond to the statement "*This song is recognisable as belonging to the musical genre*". A Likert scale, consisting of 'Strongly agree', 'Agree', 'Neutral', 'Disagree' and 'Strongly disagree', was provided. The values assigned to each possible answer are 5, 4, 3, 2 and 1 respectively.

Table 4.2: Recognisability of generated songs to their intended genre

Genre	N*	Classification** (per cent)					Mean	Standard Deviation
		SD	*D*	*N*	*A*	*SA*		
Both	122	9	25	12	45	9	3.20	1.178
Bossa Nova	62	6	27	18	42	6	3.15	1.099
Trance	60	12	23	7	48	12	3.25	1.260

* Number of evaluations
** SD=Strongly disagree; D=Disagree; N=Neutral, A=Agree; SA=Strongly agree

To the auditioning of songs within the Bossa Nova genre, forty eight percent of participants responded positively ('Strongly agree' or 'Agree') and thirty three percent negatively ('Disagree' or 'Strongly Disagree'). An improved response to the Trance genre was evident by the sixty percent supplying a positive answer and only thirty five percent providing a negative response. It is interesting to note, in Table 4.2, that few participants were neutral in their opinion on whether or not the song auditioned was recognisable to the genre.

The song BN01, achieved the highest mean for the Bossa Nova songs, with a mean of 3.71. The statement, above, received only 'Agree' and 'Neutral' values by participants. Two songs within the Trance genre attained a mean of 4.25. Participants, evaluating these songs, responded to the statement with either a 'Strongly agree' or 'Agree'. The limited range of responses indicates a consensus in opinions between the different participant. (small standard deviation in many songs).

In evaluating the success of each composition it is necessary to determine how evident the song's non-human origin is. Students responded to the statement "*This song exhibits a similar quality to that of a human-composed MIDI file*" by selecting the desired sentiment from the Likert scale provided.

Table 4.3: Participants' discernment of the song's non-human origin

Genre	N*	Classification** (per cent)					Mean	Standard Deviation
		SD	*D*	*N*	*A*	*SA*		
Both	118	10	27	16	26	20	3.20	1.316
Bossa Nova	58	6	40	16	19	21	3.09	1.302
Trance	60	13	17	17	33	20	3.30	1.331

* Number of evaluations
** SD=Strongly disagree; D=Disagree; N=Neutral, A=Agree; SA=Strongly agree

An equal number of participants provided a 'Strongly agree' response to songs from the Bossa Nova and Trance genre. Almost forty percent chose 'Disagree' to this statement in regard to Bossa Nova, which is higher then the combined total of 'Disagree' and 'Strongly disagree' responses assigned to the Trance genre.

A value of 3.50 was the highest mean response to this question for Bossa Nova songs. The question for this song received a high standard deviation value indicating contrasting opinions amongst participants. The highest mean response achieved by a Trance song was 4.00. This question achieved a distinctly low correlation with all other questions. Possible reasons for this are students formed different interpretations to the question or that a less tangible, highly subjective quality was being assessed.

The last, and arguably most important, aspect of the project's objective is the aesthetical ratings provided by the participants. The last question in the 'song evaluation' questionnaire elicits a personal rating of the song auditioned. The student is again required to respond, through the Likert scale, to the statement "*This song is aesthetically pleasing to the ear*". The results are shown below in Table 4.4.

Table 4.4: Aesthetical ratings awarded by participants

Genre	N*	Classification** (per cent)					Mean	Standard Deviation
		SD	*D*	*N*	*A*	*SA*		
Both	130	15	24	22	34	5	2.93	1.179
Bossa Nova	65	15	22	28	34	0	2.83	1.084
Trance	65	14	26	15	34	11	3.02	1.269

* Number of evaluations
** SD=Strongly disagree; D=Disagree; N=Neutral, A=Agree; SA=Strongly agree

The most frequent response was 'Agree'. This was selected by more than a third of participants in both genres. There was no Bossa Nova song which received a 'Strongly agree' response. Mean ratings for each of the Bossa Nova songs ranged between a low 2.25 and 3.43.

Trance songs fared little better. Mean values ranged between an even lower 2.22 and 3.78. Four of the seven Trance songs auditioned did not receive any 'Strongly agree' responses for this question. Only one song did not receive a 'Strongly disagree' response.

The above results indicate that songs generated within the Trance genre better accomplished the objective of automated composition. In all aspects of the objective, a better mean measure was achieved by the Trance genre. Trance songs attracted higher, positive values. However they also attracted more 'Strongly disagree' values than songs composed in the Bossa Nova genre.

The aesthetic ratings were the lowest of the measures discussed. The mean value for the combined genres was of a negative nature. The mode for each of the three questions from the 'song evaluation' questionnaire was 'Agree' in both genres, except in one case. This case was the response to the statement "*This song exhibits a similar quality to that of a human-composed MIDI file*" in songs from the Bossa Nova style. The majority of participants responding to this case selected 'Disagree'.

The overall quality assigned both genres was 'Amateur'. Participants who chose to provide an answer for each genre collectively placed a better rating on the Trance genre. This sentiment was reinforced by comments provided by participants in the 'overall evaluation' questionnaire. Such as:

"*The software worked reasonably well for the trance...*";

"*...the trance ones in general seemed to be more realistic in terms of the genre*";

"*...the trance really works*";

"*...the trance was great, bossa is iffy!*".

4.3 Compositional Aid for Musicians

A secondary objective of this project was that the software be of use to musicians in composing music. The second and third question in the 'overall evaluation' questionnaire addressed this objective directly. The participant was asked if the software would be of help in composing songs; examples of how it may be used were provided. If the student answered that it would be occasionally or frequently helpful they were asked to describe how.

Table 4.5: Value of the software as a compositional aid

Genre	N*	Classification (per cent)			Mean
		Not helpful (3)	*Occasionally helpful (2)*	*Frequently helpful (1)*	
Both	12	42	33	25	2.17

* Number of evaluations

Two of the fourteen respondents specified a different rating for each genre. The results, summarised in Table 4.5, are similar to what would be expected when considering the results from the previous question. Two trends are noticeable in question three ("*If you answered* b. *or* c. *to question 2., please explain how this software would be of help to you*"). From the six participants who specified it would be occasionally, or frequently helpful, and provided an answer to question three, two responses indicated the software would only be of use for composing Trance songs. Also, four of these six people responded that the software will be useful for "*backing tracks*".

4.4 Generation and Modification of Note Sequences

The project research questions focus on the specifics of achieving the above objectives. The following section addresses the success of the first and second research questions; "*Can sequences of notes, generated by finite state automata, reflect the characteristics found in the sample data used to create the finite state automata?*", and "*Can sequences of notes be affected through morphing, mutation, regeneration, crossover, or a combination of these methods, to accurately represent variations in note sequences typical to the genre selected?*".

The translation of these questions into a format comprehensible to the students does not allow each question to be assessed individually. The 'song evaluation' questionnaire contains a number of questions which evaluate musical aspects of the conformance to the genre by note sequences. The results from these questions are presented in the following sections.

4.4.1 Chord Voicing

Chord voicing refers to "the way the notes are arranged" (Morangelli, 1999) within a chord. The results of this question will indicate the success in the arrangement and selection of notes played simultaneously.

Table 4.6: Assessment of chord voicing in both genres

Genre	N*	Classification** (per cent)					Mean	Standard Deviation
		SD	*D*	*N*	*A*	*SA*		
Both	122	6	25	20	43	5	3.16	1.047
Bossa Nova	62	8	23	23	44	3	3.11	1.057
Trance	60	3	28	18	43	7	3.22	1.043

* Number of evaluations
** SD=Strongly disagree; D=Disagree; N=Neutral, A=Agree; SA=Strongly agree

The results are similar for both the Trance and Bossa Nova styles. For both the highest and second highest mean value achieved by a Bossa Nova song, participants selected either 'Neutral' or 'Agree'. Three Trance songs received an unfavourable mean as compared to only two Bossa Nova songs.

For most genres, the most frequent response was 'Agree', yet comments provided indicate flaws in the implementation of the chord structures. "*Melodic choices.....at times non-sensical in relation to the chord structure*" and "*There is no coherent chord progressions*" were directed at Bossa Nova songs by one participant.

The sentiment echoed by a number of students was that the chord progressions were not coherent and that the solo and melody did not fit over this progression. Verbal feedback brought the clarification that musical phrases did not conform to the relationship with chord structures that is typical to the Bossa Nova genre. The result of this, as expressed repeatedly by one individual, is "*too many clashes in harmony*".

The Trance genre was also perceived as having harmonic clashes due to "*major over minor*" occurrences. These comments signify a flaw in the implementation of the chord progressions, or a harmonic relationship between chord and melody not learnt or violated. The severity of this error is more apparent in songs created within the Bossa Nova genre.

4.4.2 Rhythmic Properties

Brazilian Jazz, such as the Samba and Bossa Nova, has a distinctive rhythmic presence. This presence is realised by the rhythm section (piano, guitar, bass, drums and percussion). A repeated clave is present through out the song. This provides a consistent rhythmic base. Rhythm within the Trance genre is also important. A driving, consistent quality is formed through the combination of all instruments.

The table below summarises the responses to the statement "*This song conforms to the genre in terms of: Rhythmic properties*". The results from this section concern not only the generation and modification of note sequences but also the interaction between different instruments.

Table 4.7: Recognisability of generated songs to their intended genre.

Genre	N*	Classification** (per cent)					Mean	Standard Deviation
		SD	D	N	A	SA		
Both	131	12	21	9	49	8	3.20	1.224
Bossa Nova	65	11	25	8	51	6	3.17	1.193
Trance	66	14	18	11	47	11	3.23	1.262

* Number of evaluations
** SD=Strongly disagree; D=Disagree; N=Neutral, A=Agree; SA=Strongly agree

The results for both the Trance and Bossa Nova songs auditioned are very similar. Both received a positive response of approximately fifty seven percent. The majority of participants agreed with the above statement when listening to Bossa Nova songs.

The highest mean achieved by a Bossa Nova song was 4.0. The responses comprised of eight, identical 'Agree' values. This consensus in student's opinions substantiates the unanimous response. A value of 4.11 is the highest mean achieved by a song within the Trance genre. Here to, there was little deviation in responses.

One of the trends identified from question three, in the 'overall evaluation' questionnaire, was that the software would be useful as a backing track. The rhythmic properties of a song would be key to its usefulness in this role. This accounts for the relatively high measures obtained from this question.

4.4.3 Musical Phrasing

Musical phrasing refers to how musical sentences are performed. The following table summarises the responses to the statement "*This song conforms to the genre in terms of: Musical phrasing*".

Table 4.8: Participants' response to musical phrasing

Genre	N*	Classification** (per cent)					Mean	Standard Deviation
		SD	*D*	*N*	*A*	*SA*		
Both	128	13	29	16	35	8	2.97	1.210
Bossa Nova	65	11	37	18	31	3	2.78	1.097
Trance	63	14	21	13	40	13	3.16	1.298

* Number of evaluations
** SD=Strongly disagree; D=Disagree; N=Neutral, A=Agree; SA=Strongly agree

The Bossa Nova genre fared poorly in this aspect. Thirty four percent of responses were positive and almost fifty percent were negative. These results are strengthened by comments provided by participants in the 'song evaluation' questionnaire.

Criticism was directed towards the following aspects of musical phrasing in Bossa Nova songs. The melody was criticised as sounding "*very random and disjointed.*" A repeated opinion was that the melody line jumps, unexpectedly, between octaves. A comment towards one of the lesser successful Bossa Nova songs was that "*phrasing of the improvising instruments is dodgy*". Poor timing, rhythm and lack of expected, chordal relationships were attributed as the cause.

Similar criticism was extended to the phrasing of the solos. The overriding sentiment was that instruments were too random in their improvisation. Ideas were not connected, tonal and rhythmic formalities not adhered to and there was no reworking of the song's motifs within the solo.

Songs generated within the Trance style received a better review. Over fifty percent of participants believed conformance to musical phrasing was achieved. The highest mean value received by a Trance song was 3.89; the lowest was 2.13.

4.4.4 Instrument Conformance

The following paragraphs discuss the outcome to the question three form the 'song evaluation' questionnaire ("*Instruments fulfil roles typical to the genre in terms of: Displaying stylistic characteristics common to the genre*"). This statement is designed to evaluate conformance to the genre from a different perspective. The results from this section concern not only the research question currently under focus, but share boundaries with other research questions.

Table 4.9: Feedback concerning instrument conformance

Genre	N*	Classification** (per cent)					Mean	Standard Deviation
		SD	*D*	*N*	*A*	*SA*		
Both	128	9	23	17	45	6	3.18	1.118
Bossa Nova	65	6	28	22	45	0	3.05	0.991
Trance	63	11	18	13	46	13	3.32	1.229

* Number of evaluations
** SD=Strongly disagree; D=Disagree; N=Neutral, A=Agree; SA=Strongly agree

As indicated by Table 4.9, in both genres, 'Agree' was by far the most frequent response. No participants strongly agreed that instruments conformed to the genre in any Bossa Nova song. The highest mean value achieved by a Bossa Nova song is 3.89, where participants either selected 'Neutral' or 'Agree'.

The results show a large contrast between the Trance and Bossa Nova genres. Fifty nine percent of respondents were positive in their response to this question when evaluating a song in the Trance style. Comparatively, only forty five percent of participants gave a favourble response to this questions for Bossa Nova songs. The mean values in the table above also reflect this difference in achievement between the genres.

As expected, there is a high correlation between answers to this question and the combined answers of the chord voicing, rhythmical properties and musical phrasing. In both genres, the rhythmic properties shared the highest correlation with this question. This indicates that rhythm was an important factor in determining the conformity of an instrument.

4.5 Instillation of Global Characteristics

The instillation of global characteristics is accomplished through the use of an instrument activity table. In the case of the Trance genre, this table is evolved while songs generated in the Bossa Nova style are from a fixed table. The research question "*Can a skeletal structure, representing typical instrument activity in a musical genre, be evolved using a genetic algorithm?*" is addressed by the following parts of the 'song evaluation' questionnaire.

Question four, "*The form of this song displays a realistic style*", addresses the primary focus of the research question. Question 3.2, "*Instruments fulfil roles typical to the genre in terms of: Interaction with other instruments*", relates to success of the instrument activity table in modelling instrument interaction. Although, the table only implements interaction at a precision of one phase (two bars). Instrument interaction within a phase relates more to the other research questions addressed in the previous section.

4.5.1 Form

The form of a song is dependent on the structure of its chord progressions and instrument dynamics. The following table summarises the results from question four in the 'song evaluation' questionnaire.

Table 4.10: Recognisability of the form of the song to its intended genre

Genre	N*	Classification** (per cent)					Mean	Standard Deviation
		SD	*D*	*N*	*A*	*SA*		
Both	128	9	32	17	34	9	3.01	1.164
Bossa Nova	63	8	36	19	33	3	2.87	1.070
Trance	65	9	28	15	34	14	3.15	1.240

* Number of evaluations
** SD=Strongly disagree; D=Disagree; N=Neutral, A=Agree; SA=Strongly agree

Songs within the Bossa Nova genre received more negative responses then positive to this statement. The majority of respondents selected 'Disagree' to songs in the Bossa Nova style. The difference between the highest and lowest mean values for Bossa Nova songs, 3.33 and 2.54, was quite small. This was expected as the instrument table was the same for each song.

Results from the Trance genre were a lot more varied. The highest mean value was 4.22 and the lowest was 2.00. The large variations between each song indicate large differences in the instrument activity tables produced by the genetic algorithm.

Success of the evolutionary process was evident in comments supplied with the 'song evaluation' questionnaire. These included "*Good fade after intro - very realistic*", "*Good dynamics*" and "*The trance ones have very good development in the form*". Within the Trance genre, there was a high correlation between the responses to this statement and those evaluating the degree of conformance achieved by instruments. This is to be expected as instruments conforming to the roles of their genre will achieve a realistic style.

4.5.2 Instrument Interaction

Interaction between instruments is only implemented at a phase level. The assumption was made that the independent combination of instrument notes sequences achieve inter-instrument interaction typical to the genre. The following table is the summary of the responses to the statement "*Instruments fulfil roles typical to the genre in terms of : Interaction with other instruments*".

Table 4.11: Instrument interaction.

Genre	N*	Classification** (per cent)					Mean	Standard Deviation
		SD	*D*	*N*	*A*	*SA*		
Both	129	7	29	27	33	5	2.99	1.042
Bossa Nova	65	6	38	26	29	0	2.78	0.944
Trance	64	8	19	28	36	9	3.20	1.101

* Number of evaluations
** SD=Strongly disagree; D=Disagree; N=Neutral, A=Agree; SA=Strongly agree

Under a third of responses to songs in the Bossa Nova style were positive. There were no 'Strongly agree' values assigned to any song. Four of the seven songs auditioned received an overall negative response for this question. The highest mean a song achieved was 3.22; the lowest 2.22.

Again, the Trance genre obtained better results. There were forty five percent positive responses compared to only twenty seven that were negative. The song achieving the highest mean value, 4.00, was the same song that achieved the highest mean measure for the form.

4.6 General

The following paragraphs address interesting trends noted in the results. These trends may help better explain the feedback obtained. First, the results to the statement "*Instrument interactivity results in a smooth, continuous flow*" are discussed.

Conclusions drawn from these results concern the project as a whole, not one specific research question or objective.

Table 4.12: Flow/fluidity

Genre	N*	Classification** (per cent)					Mean	Standard Deviation
		SD	*D*	*N*	*A*	*SA*		
Both	130	12	28	21	34	5	2.93	1.142
Bossa Nova	65	11	35	25	29	0	2.72	1.008
Trance	65	12	22	17	38	11	3.14	1.236

* Number of evaluations
** SD=Strongly disagree; D=Disagree; N=Neutral, A=Agree; SA=Strongly agree

Twenty nine percent of responses to the Bossa Nova songs were positive for this statement. The most popular selection was 'Disagree'. The achievement of this aspect of the composition was the least successful. The mean value, 2.72, is the lowest mean value allocated to any of the questions within the 'song evaluation' genre.

In relation to the other questions, songs from the Trance genre also performed quite poorly. The mean value, 3.14, was the second lowest measure collected. Although, almost fifty percent of participants provided a positive response.

The response to this statement bears the strongest correlation with how aesthetically pleasing the song was. This relation is stronger than that found between the technical achievements of the song (Q1, Q2.1-3, Q3.1-2 and Q4) and its aesthetical pleasing quality. This second correlation was expected to be higher.

Songs holding a relatively high average mean response to the technical questions often received a low mean aesthetical value, and vice versa. For example, song BN03 received an average mean value (the average of the mean responses to Q1-Q4) of 3.48 and a much lower aesthetical value of 2.56. There was, how ever, a high correlation between the technical average of a song and its recognisability to the genre.

For each genre, there was one song which performed distinctly better than the rest. This one song holds the best performance for most of the questions. This is expected as there are not distinct boundaries between the song characteristics each question addresses.

There was little correlation between question six "*This song exhibits a similar quality to that of a human-composed MIDI file*" and any other question. This may indicate that participants had different interpretations and ideas about what constitutes a non-human composed song. The highest correlation found was between question six and question five "*Instrument interactivity results in a smooth, continuous flow*".

A frequent comment attracted by the Trance style songs was that the voices of the instruments were characteristic to those used in music of the 1980s. This was an expected sentiment. The General MIDI instruments used to audition the generated songs sound similar to the synthesisers frequently used during the 1980s.

4.7 Summary

The most evident trend observed in these results is the difference in accomplishment of the objectives by both genres. Songs from the Bossa Nova genre consistently achieved a measure inferior to those achieved by songs composed in the Trance style. This view was reinforced by the comments and feedback provided by the participants.

Specifically, results relating to the instrument interaction, the musical phrasing, fluidity of the piece and evidence of a human origin performed poorly. The overall aesthetical rating for the Bossa Nova genre was negative. There were three of the seven songs that averaged a rating which was not negative, but none received any 'Strongly

agree' responses for this aspect. The general classification of Bossa Nova songs was below amateur.

Despite these unpromising results, one song, from the seven consecutively generated, achieved a relatively decent appraisal. Rhythmic properties and recognisability to the genre were accomplished reasonably well overall and very well in the song mentioned. This interpretation is strengthened by the participant's responses indicating these songs would be useful as a backing track.

The mode indicated that most participants selected 'Agree' as their response to the statement "*This song is aesthetically pleasing to the ear*". Although, no participant responded with 'Strongly agree' to this statement. The highest correlations to the aesthetical responses was found in the responses for musical phrasing and fluidity of the song. As expected, a high correlation exists between the rhythmic properties of the piece and its recognisability to the genre.

Songs generated within the Trance style achieved a greater level of success in accomplishing the objectives. In both quality and usefulness, when addressed separately by respondents, the Trance genre always obtained a higher rating. This was evident in both the responses to the 'overall evaluation' questionnaire and comments directed to songs in the 'song evaluation' questionnaire.

Fifty nine percent of students provided a positive response concerning the song's recognisability to the genre and the instrument's conformance to the genre. All questions within the 'song evaluation' questionnaire received the strongest, positive response, 'Strongly agree'. The mode response to each question within the 'song evaluation' questionnaire was 'Agree'. The mean response was always positive.

This differed to the mean responses obtained for the Bossa Nova songs in which as many positive values were evaluated as negative ones. The aspect of the Trance songs that performed the least well was the aesthetic ratings. This was probably partly due to nonconforming, eighties sounds produced by the general MIDI instruments.

A song achieving high scores in all technical aspects did not imply a high aesthetical rating. The fluidity of the song had the strongest correlation with the aesthetical rating assigned. Both genres fared poorly when assessed on their fluidity. A low correlation exists between a song's recognisability to the genre and its aesthetical value. A song's rhythmic properties shares a high correlation to its recognisability to the genre.

5.0 - DISCUSSION

5.1 Obtaining an Objective Critique

To obtain an accurate, objective evaluation of the generated compositions was problematic. The main difficulty was the auditioning of the internal representation of the song without inducing external bias. The necessary process of translating the internal representation into a machine-readable format involves alterations and decisions that significantly effect the resulting music auditioned.

The format of MIDI was chosen for several reasons, the universality and portability of the format, the student's familiarity with the format, and the availability and accessibility to songs of in this format. The internal representation of the song consists only of notes; translation between this representation and MIDI requires instruments to be assigned.

Instruments were selected from the standard, general MIDI list of 128 instruments. This list is very restrictive, especially when selecting instruments for the Trance genre. Comments were received expressing the need for more 'modern' sounds and less 1980s'ish instruments.

Different attack and decay settings inherent to each instrument effected the way notes were played. The duration specified by the internal representation of the song was often not adhered to by the instrument. Sample data obtained was instrument dependent.

The adjustment in instrument settings, such as reverb, echo and delay, was judged to be outside the scope of the project. Thus instruments selected were left in their default state. This effected a raw, unpolished quality in the resulting songs. Manual assignment of instrument settings to those common to the genre may have improved the aesthetical quality of songs generated but was considered to detract from validity of the translation.

The process of translating, as faithfully as possible, the internal representation of the song enabled the auditioning of a song for evaluative purposes. A survey was conducted using Western Australia Academy of Performing Arts (WAAPA) students as the target population. Limited data was obtained due to the regular problems associated with recruitment. These problems were compounded by the time required of participants in evaluating a large number of songs.

The design of a survey which addressed the research questions and technical details of the software process was difficult. A compromise between musical comprehensibility and the distinct address of the technical aspect requiring assessment was needed. Certain technical features of the software could not be individually addressed. The resulting questions shared blurred boundaries and relationships with each other.

Fifteen students participated and fourteen different songs were auditioned, seven from each genre. The duration of each song was approximately three minutes. On average, students evaluated around nine songs each. Despite the problems encountered and the relatively small amount of data collected valuable, written and verbal feedback was obtained. The implications drawn from this feedback are discussed below.

5.2 Note Sequence Generation and Modification

Analysis of the results of the survey indicated a failing in the software's approach to chord voicing and melodic phrasing within the Bossa Nova genre and, to a smaller degree, in the Trance genre. The process of transposing note sequences generated from the unformatted source data is flawed. Feedback indicates that chord sequences are not accurate to the genre and, at times, not even discernable.

The process of producing simultaneous notes (chord voicings) is identical to that of generating a sequence of non-simultaneous notes. This means the notes, generated by the finite state automata, may be obtained from a chord, a melodic phrase or both. The arrangement of notes within a chord, especially within Jazz, is a complex procedure. Simultaneous notes need to be addressed by a different process than the process generating a melodic phrase.

Results also point to a lack of conformance, by the improvising instrument, to the associated theory. The software process did not learn, or did not retain, the relationship between the melody and the chord sequence. This relationship may have been destroyed by the transposition of the generated note sequence to the specified chord.

The transposition of finite state automata generated note sequences has also had the adverse effect of creating undesired jumps in the melody. The method of generating a melodic sequence, transposing this sequence, then placing it within the continuing song is unnatural. This sequence is probably one of the main causes of unfavourable

feedback regarding this aspect of the song. Another possible cause, discussed later, is the multiple, independent generation, and combination, required to form one melodic phrase.

The preliminary assumption, note attributes, independently generated from the same training data, will combine to produce note sequences representative of that training data, appears to be valid. A definite gauge of the validity is difficult as the assumption is not directly addressed by any one question.

Verbal and written feedback, in addition to responses to questions indirectly related, provide an indication. These results suggest failings in the dependency between instrument note sequences. Negative criticism was not directed at the musical phrases, outside of the context of the song, in relation to their conformance to the genre. However, this may be because to do so is an unnatural way to express the deficiency.

Unfortunately, due to time restrictions, the effect of varying the amount of sample data used was not explored. It is expected elements whose attribute sequences are inherently diverse will require a larger amount of sample data to accurately model the genre. It is possible that an increase in sample data used by improvising instruments will result in a more positive reception by survey participants.

5.3 Global Characteristics

The form and instrument interaction found within a song, generated in the Bossa Nova style, was poorly reviewed. Feedback directed at songs composed in the Trance genre were more positive. Songs generated in the Trance style received more variability in their ratings, achieving both the highest and lowest evaluation for this aspect.

The template used to generate Bossa Nova songs is a predetermined instrument activity table. The form encoded by this table is one that is familiar to those found in the sample songs. The instrument activity table partitions the song into equal length phases. One phase is of two bar length. This means instrument activity within two bars can not be explicitly modelled.

A project assumption was that instrument interaction, within the two bars, would be achieved through the independent combination of the instrument note sequences. Verbal and written responses indicated that this assumption was not valid. The nature of Jazz is one of spontaneous interaction and improvisation. Performers, represented by elements, feed off each other creating a strong dependence between various instruments.

The consequences of this incorrect assumption was partially negated by the clave. This repetitive, two bar rhythm provides an similarity between distinct, rhythmic structures formed by the finite state automata of different elements. This similarity occurs because instruments share a dependency to this base, consistent clave in addition to other instrument's current disposition.

Further difficulties were caused by the partitioning of the song in to two bars phases. The formatting of melodic phrases, obtained for input data, involved fragmenting the phrase into two bar segments. This often involved the truncation of notes ending after the two bar limit. The post formatted collection of two bar segments retain little of the original structure of the melodic phrase.

Consequent musical phrases are formed from multiple, two bar segments. The combination and transposition of these segments results in the problems discussed earlier. A more natural approach would be to employ a varying length phase equal to one melodic phrase. This would help keep the phrase's structure intact. Also, the relationship between the phrase and the chord sequence may be better preserved.

The Trance genre does not suffer the same weaknesses as those described above. The two bar segmentation is musically sensible for songs composed in the Trance style. This is because the majority of melodic phrases are aptly contained within the two bars. No truncation or combination is necessary when formatting or forming a melodic phrase within the Trance style.

The assumption that instrument interaction can be achieved through the independent combination of the instrument note sequences appears to be valid in the context of the Trance genre. Trance instruments combine to produce the rhythmic structure. This differs to Bossa Nova songs, where instruments respond to each other. The instrument interaction in Trance is significantly less complex than in a Jazz.

Results indicate that the evolutionary process of evolving the instrument activity table had varying amounts of success. Of the Trance style songs generated, some received unanimous, positive criticism and others only negative. This is due to the unpredictability of how the genetic algorithm will satisfy the search criteria; the personal, subjective views of what features the form a Trance song must exhibit; the instrument interaction formed within each phase.

5.4 Fluidity

A high correlation is present between the aesthetical value of a song and the fluidness this song achieved. The fluidity of a song also shared a high correlation with the evidence of the song's non-human origin. Survey results indicate that songs from both genres performed relatively poorly in this aspect. Songs from the Bossa Nova genre were perceived as regimented and the average smoothness rating was the lowest rating received.

The high correlation between the song's origin and its fluidity is predictable. Erratic connotations are associated with machine-generated music. Conversely, a human composed song is expected to flow. The strongest correlation of a song's aesthetical quality is to its fluidity was less anticipated. A survey conducted amongst a larger population may produce different correlations.

The poor performance, by both genres, in achieving a smooth, continuous flow can be attributed to a number of factors. These include the formatting of the source note attribute sequences obtained, the problems associated with translating the project's representation of the song into a machine-readable format, and the segmentation of musical phrases in the Bossa Nova genre, discussed previously. Another major factor is the unpredictability of the evolution of a solution activity table.

Data note attribute sequences undergo a formatting process which rounds the note attribute values. Position and duration values are rounded to the nearest thirty second note and velocity values are rounded to the nearest number divisible by four. The purpose behind the rounding is to facilitate a greater number of merges between states in the creation of the finite state automata. This enables the finite state automata to produce greater varieties and more unique sequences of note attributes.

The effect of rounding the note attributes is minimal in note sequences obtained for the Trance genre. It is rare that an audible loss of accuracy will occur from the rounding process. This is due to the relative simplicity, high tempo and machine-like precision of Trance music.

Note sequences from the Brazilian Bossa Nova music, a sub genre of Jazz, display a human-like characteristic through a subtle variations in note placing (positions), velocity and duration. The result of the rounding process is an audible loss in rhythmic feel. This is particularly evident in the solo sequence, as stated by a number of participants.

5.5 Future Considerations

The songs auditioned in the survey were the first seven songs generated in each genre. Some failings identified by participants of the survey may not appear in subsequent generations of songs. From this perspective, the project's flaw may be classified based on the technical feasibility of achieving a 'correct' solution with the current software.

The following are aspects of the project that require an improvement or correction. The project lacks the means to produce the desired effect within the song. Most of these failings are more evident in the Bossa Nova genre. The project's approach to song composition is more amenable to the Trance genre than the Bossa Nova genre.

The independent combination of instrument note sequences does not effectively model the intricacies of inter-instrument dependence. The software process does not learn, or instil, the important instrument interaction that is typical to the selected genre. For this reason, the probability of the project, in its present state of generating a song displaying these neglected characteristics is small.

Two possible approaches to instilling the inter-instrument dependencies are considered. A rule base, derived from theory associated with the genre, may be applied to generated note sequences. Alternatively, the interaction may be learnt from supplied MIDI files. Both approaches require that note sequences are formed with respect to note sequences already existing within the phase.

The implementation of the chord progression is also deficient. Survey results indicate that the transposition of note sequences into the desired chord voicing is not accurate. Further, the application of the chord to the note sequences has the undesirable consequence of destroying chromatic passing notes not part of the scale.

The need of a separate method for generating chord voicings has been mentioned. A different representation for the simultaneous notes must be used consisting of; the number of notes within the chord, the position of each note in relation to the chord. Note arrangements of chords within the sample songs must be learnt and this information used to generate resulting chords.

The process of applying the scale to the largely unformatted note sequence is flawed. The formatting of note sequences obtained from sample songs may require their separation based on tonality. The data used in the generation of a note sequence, by an

element, will be dependent on the associated chord. Resultant application of the scale will be less intrusive.

Vital relationships between melodic phrasing, or improvisation, and the chord sequence are not present. This may be due to the inaccuracy of the note sequence transpositions, the segmentation and subsequent combination of melodic phrases and an inability to learn this relationship from the source data.

The formatting of the source data, and the eventual translation of the song into a machine readable format, results in a loss of fluidity and human-like subtlety.
This problem can be addressed in two different ways.

The precise state merging conditions the finite state automata utilise may be altered to allow more flexibility. A merge between two states occurs only if there is an exact match for the state's attribute and this attribute's position within its original note sequence. Note attributes are rounded to increase the number of possible state merges. Alternatively, a match between state attributes may be more lenient by merging states containing attribute values that are within a certain range.

A second approach is to reapply the subtle variations. This can occur after the generation of the note sequence by the finite state automata. The deviation from the current, rounded figures may be derived from the source data. For example, the difference between a new, generated note attribute sequence and its rounded counterpart will provide the current note sequence with values to create a subtle variation typical to the source data.

The form of a song generated in the Bossa Nova style was considered to be lacking. The inflexibility of the instrument activity table, due to time restrictions, limited the potential success of the song. Also, the two bar precision assigned to one phase restricted the softwares capability to specify instrument interaction within the two bars.

Further, this stipulation necessitates the separate generation and combination of note sequences to form musical phrases exceeding two bars in length. The manual forming of a phrase was not successful and, possibly, causes additional problems.
A previous, suggested solution was the use of a varying length phase that accommodates the length of the musical phrase. This is only a partial solution - instrument interaction within a phase must also be addressed.

Two distinct directions in improving the software are instilling rules based on theory associated with the genre or learning the necessary characteristics through the sample songs provided. The second approach is more desirable. This approach provides a flexibility in the genres, and subgenres, that the software is capable of modelling.

This approach also has its related disadvantages. A difficulty exists in selecting a source format from which extracting and learning the required information possible. A difference in the variety and significance of song characteristics between genres will create a demand for a highly flexible process. Further, an increased amount of sample data exhibiting all necessary characteristics will be required.

5.6 Accomplishment of Objectives

Based on results from the survey, songs generated in the Bossa Nova style contain fundamental shortcomings. Bossa Nova songs were conveyed by participants to be amateur. The overall, aesthetical response to these songs was negative. With out modification the software is unable to produce an aesthetically pleasing Bossa Nova song that conforms to all the genre's complexities.

This said, one of the seven songs achieved a decent review. Questions relating to the software's deficiencies were answered positively overall. This may be attributed to the possibility that shortcomings were not as evident within this song. The diversity of responses indicate a varying ability, by participants, to perceive these shortcomings.

In contrast to Trance songs, a Bossa Nova song must exhibit a warm, human-like quality. Both the translation to MIDI and the auditioning by MIDI instruments detract greatly from the song's potentiality. As stated by one participant "*Bossas are played better by humans*". Consequently, a live band may elicit a more accurate evaluation of the generated song.

The primary objective of providing non-skilled people the ability to generate songs in the Bossa Nova genre was not successfully met. Fundamental deficiencies, though not distinctly obvious to the untutored ear, prevent the generated songs from being technically proficient. The average rating of a generated songs is at best amateur, restricting the possibility of any commercial use.

The secondary objective of assisting musicians in the composition of music was more successfully accomplished. This was indicated by the responses addressing this intention in the survey. Further, shortcomings identified largely affect the relationship

and dependencies between instruments and chords. As a compositional aid, generated note sequences will be individually useful.

Survey results indicate the Trance genre was more successful in achieving both objectives. The majority of songs generated attracted a positive, aesthetical rating. The most successful song, generated in the Trance style, attained an average response between 'Agree' and 'Strongly agree' for the majority of questions. Positive, written feedback was directed to many aspects of songs generated in the Trance style.

Several reasons are attributed to why the shortcomings identified in the Bossa Nova genre were less apparent in the Trance genre. These were; the software's approach lent itself to the generation of Trance style songs, the relative simplicity of Trance songs, especially in areas where the software underachieved, and the machine-like precision of note placement typical to the Trance genre.

The translation into MIDI did not adversely affect the song's evaluation to a great extent as the effect on Bossa Nova songs. However, the audition of the song by the restrictive, GM instruments probably limited the ratings the song might have achieved. Songs were considered, by participants, to resemble music from the 1980's. It was expressed that songs generated in the Trance style "*need more modern sounds*".

The primary objective of automated composition was successful to a degree. The software was able to generate music that conformed to the specifications set. That is, not obviously machine generated, recognisable to the genre, aesthetically pleasing, and usable in a commercial context. The latter requirement not being directly tested.

Survey comments indicated Trance songs were "*realistic*" and recognisable to the genre. It is anticipated the aesthetical rating will increase if auditioned with superior, genre-specific instruments. The assumption is made that a high conformance to the previous requirements will imply usability in a commercial sense.

The secondary goal of assisting a musician in composing is also considered to be moderately successful. A positive classification of the quality of the Trance songs, by the participant, did not always imply they would find it useful. It is assumed that the participant would not find it useful due to a general aversion to using compositional aids or the unlikeliness of composing a Trance song.

An accurate assessment of the research questions is difficult due to the indirect nature of the survey questions evaluating them. This was especially the case with the research question "*Can sequences of notes, generated by finite state automata, reflect the characteristics found in the sample data used to create the finite state automata?*".

However, in both genres it seemed note sequences accurately reflected the characteristics found in the sample data. The rhythmic elements of the Bossa Nova songs underwent little, or no, modification. Comments and question responses pertaining to these elements were generally positive. Overall conformance to the genre, by the Trance instruments, was designated as being very high. Problems arose from the lack of interaction between the note sequences rather than the note sequences themselves.

A more distinctive response was provided for the research question "*Can sequences of notes be affected through morphing, mutation, regeneration, crossover, or a combination of these methods, to accurately represent variations in note sequences typical to the genre selected?*". Improvising instruments within the Bossa Nova genre was not received well.

The simple technique of selective regeneration did not effectively model an improvising instrument. Regarding all improvising instruments, but in particular the soloing piano, the overriding message conveyed was that the phrasing was too random. Theory was not conformed to, ideas were not connected, and variation on the melodic phrases were not incorporated.

The elements in Trance songs undergoing mutation or morphing did not attract any negative comments. The high appraisal of instrument conformance holds to this research question too. Unfortunately, there was limited time for the experimentation and evaluation of note sequence modification techniques.

The final research question "*Can a skeletal structure, representing typical instrument activity in a musical genre, be evolved using a genetic algorithm?*" only applied to the Trance genre. The assumption made that "two bars will provide enough precision to accurately model instrument activity" is invalid in the context of the Bossa Nova genre.

The evolution of the instrument activity table, for the Trance genre, had varying levels of success. Trance songs, generated using this procedure, received both the highest and lowest ratings assigned relating to global structure. It is expected that

further time spent on refining the set of evaluation criteria, through trial and feedback, will improve the consistency and peak performance of the genetic algorithm.

There was reasonable success in the achievement of the research goals. Unfortunately, this success was not enough to ensure an accomplished piece of music. Dependencies and relationships, not accounted for by the project, and translation inconsistencies are attributed to the shortcomings of the software. The evidence of these failings are less apparent in the Trance genre but integral in the failure of the Bossa Nova genre to complete the primary objective.

6.0 - CONCLUSION

The stated, primary objective was "to create software that will give non-musical people the ability to compose convincingly 'real' music in two contrasting musical genres". The definition of 'real' was music "which is not obviously 'machine generated', recognisable as being of the selected genre, of an aesthetically pleasing quality (a very subjective concept), and usable in a commercial context.".

To determine the degree of success the software achieved this objective a survey was conducted. The survey involved fifteen Jazz students currently attending the WAAPA. Fourteen songs were auditioned to the students who were required to evaluate the technical and aesthetical properties of the generated songs via the completion of a questionnaire.

Seven songs from each genre were auditioned. To obtain a more accurate gauge of the software's capability, the first consecutive seven songs generated were used in the survey. The translation, required to audition the internal representation of the song, was rendered as faithfully as possible into a machine playable format.

Certain songs auditioned were regarded by participants as being of human origin, recognisable to the intended genre and aesthetically pleasing. The question of whether these songs were usable in a commercial context is highly dependent on a number of factors; the musical ability of the intended audience, the format the song will be auditioned in, and the necessity that generated songs consistently achieve a commercial quality.

Despite the success of certain songs generated, general feedback indicated severe deficiencies in the technical proficiency of the compositional process. These shortcomings significantly effect songs generated in the Bossa Nova style. The software fails to produce Bossa Nova songs of a professional quality and is therefore deemed to have failed the primary objective.

The secondary objective of the project was to provide musicians with a useful, automated compositional aid. Eight of the twelve students who responded to the question specified they would find the software occasionally or frequently useful as a compositional aid. This indicates a potential use as a musicians aid and completes the secondary objective.

Research questions, detailing the realization of said objectives, were accomplished moderately well. However, flaws occurred in aspects of the compositional process that were not addressed in the design or not accurately achieved in the implementation.

Questionnaires designed were not able to directly address each research question due to the difficulty in translation between technical and musical terminology.
An indication of the success achieved are derived from personal observation, written and informal survey feedback, and a combination of responses to questions indirectly addressing the research question.

Note sequences generated from finite state automata appear to accurately reflect characteristics found in the sample data. This implies the assumption that generated note attribute sequences may be independently combined is valid. Modification techniques utilised in modelling improvising instruments was unsatisfactory. Note modification techniques applied to other instruments appeared to achieve a greater degree of success.

A skeletal structure was evolved that represented instrument interaction typical to the Trance style. Evolution of the skeletal structure was not attempted in the generation of Bossa Nova songs. The assumption that a phase length of two bars provides precision enough to model instrument interaction is invalid in the context of the Bossa Nova genre.

The deficiencies in the compositional process are attributed to songs from both genres but manifests itself more evidently in the Bossa Nova genre. Chord sequences are often indistinct and the formation of chords uncharacteristic of the genre.
The melodic phrasing is disjointed and its relationship to the chord sequence untrue to the genre. Typical, intra-phase instrument interaction is lacking.

Difficulties were encountered in the process of obtaining an accurate, objective critique of the songs generated. The conversion of the song's internal representation to a machine readable format affected a slight imprecision and negative bias from the format's limitations. This was evident from the written comments acquired.

The invariably problematic attempt in obtaining enough data was compounded by the time required to evaluate a song. As a result, only a limited amount of data was obtained. Supported by informal and written comments and personal appraisals, it is assumed trends identified provide an accurate assessment.

An interesting detail derived from the survey is the low correlation between the technical proficiency of a song and its aesthetical rating. An expected, high correlation exists between the recognisability of the song and the rating assigned to its rhythmic properties.

The aspect of the song holding the strongest correlation to the aesthetical rating is its fluidity. Songs from both genres fared poorly in achieving a smooth, continuous flow. This is attributed to MIDI restrictions and the formatting of the obtained data.

The most significant finding from the survey is the superior accomplishment of songs generated in the Trance genre when compared to those generated in the Bossa Nova style. This distinct contrast is evident in both the responses to questions, verbal feedback and written comments.

The enhanced success of the Trance genre is accredited to several, primary reasons. The approach to the compositional process lends itself to the generation of Trance songs. The complexity of the requirements for modelling a Bossa Nova song far outweigh those necessary in modelling a Trance styled song. This is especially the case in aspects of the software process that are inadequate.

On reflection, the selection of the warm, human-oriented Bossa Nova genre was a difficult challenge to model. The machine based translation and auditioning detracted from the potentially positive feedback it may have received. However, fundamental flaws, evident in the generated pieces, exist in the compositional process that prevent a Bossa Nova song achieving the primary objective.

The auditioning of Trance songs with enhanced, genre-specific instruments will have a definite, positive effect. The assumption has been stated that high ratings imply usability in a commercial context. Based on this assumption, it is anticipated a song generated in the Trance style will achieve the primary objective.

7.0 - REFERENCES

Alberti, M., Marelli, P., & Sabadini, N. (1993). Automata and NeuralWorks. *Proceedings of the ICTE 93.* Boston: MA.

Alpern, A. (1995). *Techniques for algorithmic composition of music.* [Online]. Available WWW: http://hamp.hampshire.edu/~adaF92/algocomp/algocomp95.html

Blackwell, T. Bentley, P. (2002). *Improvised music with swarms.* Department of Computer Science, University College London, Gower St., London, UK

Bod, R. (2001). *Probabilistic grammars for music.* ILLC, University of Amsterdam, Nieuwe Achtergracht 166, 1018 WV Amsterdam

Brown, B. R. (n.d.). *Introducing algorithmic composition.* Queensland University of Technology Victoria Park Road, Kelvin Grove 4059, Australia

Chen, C. Miikkulainen, R. (2001). *Creating melodies with evolving recurrent neural networks.* Department of Computer Sciences, University of Texas at Austin, TX 78712, Austin

Clement, B. (1998). *Learning harmonic progression using markov models.* [Online]. Available WWW: http://ai.eecs.umich.edu/people/bradc/papers/545proj.ps

Collins, N. (2000). *Algorithmic composition methods for breakbeat science.* Centre for Electronic Arts, Middlesex University, United Kingdom

Furstner, M. (2000). *Jazz improvisation course.* [Online]. Available WWW: http://www.jazclass.aust.com/im1.htm

Henz, M. Lauer, S. Zimmermann, D. (n.d.). *COMPOSzE : Intention-based music composition through constraint programming.* University of Saarland, Im Stadtwald, D-66041 Saarbrucken, Germany

Hirzel, M. Soukup, D. (2000). *Natural language processing.* [Online]. Available WWW: http://www-ugrad.cs.colorado.edu/~hirzel/papers/nlp00-viterbi-jazz.pdf

Järveläinen, H. (2000). *Algorithmic musical composition.* Telecommunications software and multimedia laboratory, Helsinki University of Technology

Luger, G. (2002). *Artificial intelligence.* Edinburgh Gate, England: Essex.

Manzolli, J. Moroni, A. Von Zuben, F. Gudwin, R. (n.d.). *Vox Populi : Evolutionary computation for music evolution.* [Online].

Available WWW: http://www.ici.org.br/invencao/papers/Manzolli.html

Morangelli, M. (1999). *Jazz, a short history.* [Online].
Available WWW:
http://www.thereelscore.com/PortfolioStuff/PDFFiles/HistoryJazz.pdf

Papadopoulos, G. Wiggins, G. (1999). *AI methods for algorithmic composition : A survey, a critical view and future prospects.*
School of Artificial Intelligence, Division of Informatics, University of Edinburgh, 80 South Bridge, Edinburgh, EH1 1HN, Scotland

Pearce, M. (2000). *Generating rhythmic patterns : A combined neural and evolutionary approach.*
Division of Informatics, University of Edinburgh

Pearce, M. Wiggins, G. (2001). *Towards a framework for the evaluation of machine compositions.*
Department of Computing, City University, Northampton Square, London EC1V OHB

8.0 - APPENDIX A: Definition of Terms

8.1 Technical

- **Finite state Automata**
  A mathematical model of a system consisting of a finite number of internal states and transition criteria between these states.
- **Genetic algorithm**
  An artificial intelligence based technique which uses evolutionary methods to evolve a solution from a population of potential solutions.
    - **Critic**
      Evaluation of the fitness of each member in a population in a genetic algorithm is performed by a critic.

8.2 Musical

- **Bar**
  The divisions of equal duration in a piece of music. Bars are used to keep track of time through out a song.
- **Bossa Nova**
  A Brazilian musical style fusing characteristics of Samba and Cool Jazz.
- **Chord**
  Three or more notes played simultaneously.
- **Chord sequence**
  A progression of chords, also referred to as a 'chord progression'.
- **Chord voicing**
  The arrangement of notes within a chord.
- **Clave**
  A rhythmic pattern spanning two bars. Forms the basis for arrangement and improvisation.
- **Form**
  The arrangement of recurring chord progressions to develop the structure of a song.
- **General MIDI**
  A standard soundset providing compatibility between manufacturers.
- **Improvisation**
  Spontaneous composition. Instant composition by a musician during performance.
- **Key**
  Defines the tonality of a song.
- **Melody**
  The focal, often singable, aspect of a song.
- **MIDI**
  Musical Instrument Digital Interface is a format that is designed for the communication of music between musical instruments and computers.

- **MIDI file**
  Musical directions, triggering the computer's synthesiser, stored in an electronic format.
- **Phrase**
  A melodic form analogous to a grammatical sentence.
- **Note**
  Smallest unit of music. A note is defined by its attributes, pitch, position, velocity, duration.
- **Octave**
  A musical distance measuring an interval of eight.
- **Scale**
  The predefined pattern determining note steps within an octave. Examples include the Major scale.
- **Tonality**
  The relationship between tones within a composition.
- **Track**
  Provides for the grouping of technical and musical aspects for one instrument within a music software application.
- **Trance**
  A beat-driven style of modern, electronic dance music.
- **Transpose**
  Transform a piece of music from one key to another.

8.3 Internal

- **Element**
  Provides for the modelling of an instrument type, such as Bass, by the software.
- **Expanded instrument activity table**
  A table defining the occurrences of an instrument type within a song.
- **Instrument activity table**
  A table defining the occurrences of an instrument group within a song.
- **Note attributes**
  - **Duration**
    Length, measured in ticks, a note is to be played.
  - **Pitch**
    Musical measure of the frequency of a note.
  - **Position**
    A measure, in ticks, of the starting position of the note within the phase (two bars).
  - **Velocity**
    A measure of how hard a note is to be 'struck'.
- **Phase**
  A set time length of two bars. Each column within the instrument activity table represents the instrument activity within one phase.
- **Ticks**
  The software application-specific measurement unit for specifying the position within a bar.

9.0 - APPENDIX B: Survey Questionnaire

QUESTIONS FOR EACH SONG

ID _____
SONG _____
GENRE [Bossa Nova][Trance] **[Please circle one]**

[**SA**: Strongly agree; **A**: Agree; **N**: Neutral; **D**: Disagree; **SD**; Strongly disagree]

[For questions 1 to 7, please tick the appropriate box]

		SA	A	N	D	SD	Unsure
1.	This song is recognisable as belonging to the musical genre.						
2.	This song conforms to the genre in terms of :						
	2.1 Chord voicing						
	2.2 Rhythmic properties						
	2.3 Musical phrasing						
3.	Instruments fulfil roles typical to the genre in terms of :						
	3.1 Interaction with other instruments.						
	3.2 Displaying stylistic characteristics common to the genre.						
4.	The form of this song displays a realistic style.						
5.	Instrument interactivity results in a smooth, continuous flow.						
6.	This song exhibits a similar quality to that of a human-composed MIDI file.						
7.	This song is aesthetically pleasing to the ear.						

8. Are there any further comments about the piece you would like to express?

_____
_____
_____
_____

10.0 - APPENDIX C: Survey Questionnaire

OVERALL EVALUATION

ID _____

1. Which one of the following classifications do you feel best describes the quality of these songs?
[Please tick only one box]

a. Novice/Beginner	
b. Amateur	
c. Professional	
d. Commercial quality	
e. Other : _____	

2. Based on the songs you have evaluated, do you think this software may be of help to you in composing songs?
[Please tick only one box]

a. It would not help me at all	
b. It would be occasionally helpful as a **compositional aid***	
c. It would frequently be helpful as a **compositional aid***	
d. Other : _____	

***Compositional Aid Examples :**
- Generating a base 'skeleton' for a song.
- Generating instrument parts such as a drum track.
- Inspiring or providing ideas for musical phrases

3. If you answered b. **or** c. **to question 2., please explain how this software would be of help to you.**

_____

_____

4. Are there any further comments you would like to add?

_____

_____

_____

_____

End of Survey

Thank you for your time in completing this survey.

11.0 - APPENDIX D: Object Oriented Overview

The result of the process is achieved, primarily, through the interaction between the midiFile object and objects derived from the element class. The midiFile class is a control class which co-ordinates the element objects to create the MIDI file.

Each class derived from class element represents an instrument type. Instances of these classes contain a fixed length of currently valid notes for their respective instrument. In response to external messages, the object is able to manage the state of its note sequence. This enables client objects to request time sensitive information.

Upon construction of an object derived from the class element, finite state automata are created from the data provided. One finite state automaton is created for each note attribute, pitch, position, duration and velocity. Note attribute sequences are independently generated from each finite state automaton and combined to initiate the object's note sequence.

The class element has a virtual function createVariation. This function is inherited by each derived class and re-implemented to achieve the note sequence modification required to model the instrument type the object is representing.

A table representing the activity of each instance class element is evolved. Each row represents an instrument type (element) and each column a fixed time length (phase). The cells within this table contain hold information indicating whether the element is active and, if it is, the type of note sequence modification required.

Class midiFile iterates through this table column by column. If the element is active, the current note sequence held by the element object is retrieved. Contents of the cell is sent, as a parameter, to the element's createVariation function which then alters the note sequence accordingly. For each cell traversed, the note sequence received is transposed to the key specified for the current phase. The transposed notes sequence is then converted into a MIDI format and stored in the appropriate track.

Class midiFile receives the activity table as a parameter to its general constructor. A blank MIDI file is then created with the appropriate number of tracks. Traversal of the table results in the conversion of the internal representation of the song into a machine readable format. Additional MIDI specific parameters are set, such as tempo and key. The file is then closed and saved to a predefined directory and filename.

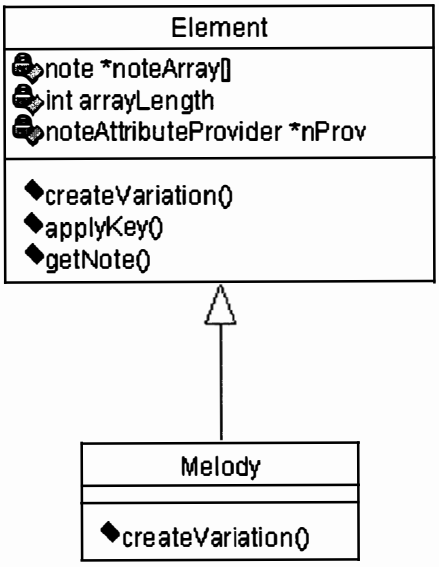

Figure 11.1: A class diagram depicting the inheritance relationship between Melody and Element